



**JAETTUJEN UNIX-TIEDOSTORESURSSIEN  
KÄYTTÖÖNOTTAMINEN  
z/OS-KÄYTTÖJÄRJESTELMÄSSÄ**

Veli-Matti Malinen

Opinnäytetyö  
Kesäkuu 2012  
Tietojenkäsittely  
Tietojärjestelmäosaaminen, YAMK

TAMPEREEN AMMATTIKORKEAKOULU  
Tampere University of Applied Sciences

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Tietojärjestelmäosaamisen suuntautumisvaihtoehto, YAMK

MALINEN, VELI-MATTI:

Jaettujen Unix-tiedostoresurssien käyttöönottoaminen z/OS-käyttöjärjestelmässä

Opinnäytetyö 70 sivua, joista liitteitä 5 sivua  
Kesäkuu 2013

---

Kelan IT-osastolla tehtiin vuonna 2010 päätös muodostaa ja ottaa käyttöön jaetut Unix-tiedostoresurssit suurtietokoneympäristössä. Nykyinen ongelma on suurtietokoneella eristetyksi toimivat Unix-ympäristöt, minkä vuoksi Unix-tiedostoresurssit järjestelmien välillä eivät ole dynaamisesti toistensa saatavilla. Tämä tekee palveluiden toimivuuden varmistamisen vaikeaksi, koska palvelut mahdollistavien ohjelmistojen kahdentaminen suurtietokoneella on hankalaa. Asiantuntijoille rajoitus tarkoittaa lisätyötä, suurtietokoneella Unix-tiedostoja käyttävän ohjelmiston ollessa käytännössä sidottu yhteen ainoaan järjestelmään. Ohjelmistojen toimivuuden sekä niiden käyttämän tiedon eheyden varmistaminen vaatii ylimääräistä tarkkaavaisuutta, jotta niitä käyttävät kriittiset palvelut olisivat varmasti asiakkaiden käytettävissä.

Opinnäytetyön tarkoitus oli kuvata toteutettu Unix-tiedostoresurssien jakaminen kohdeympäristössä sekä esitellä muutoksen mukanaan tuomia kehittymismahdollisuuksia. Ainutlaatuisesta ympäristöstä johtuen opinnäytetyö on case-tutkimustyyppinen. Raportissa selvitetään kohdeympäristössä käytettävät järjestelmät, niiden erityispiirteet ja rajoitukset. Lisäksi siinä pohditaan näiden järjestelmäratkaisujen tehostamiskeinoja ja konkreettisia tapoja, joilla kohdeympäristön päällä toimivien palveluiden toimintavarmuutta voitaisiin nostaa.

Opinnäytetyö tarjoaa kehittymismahdollisuuksia, joiden avulla Kelan suurtietokoneen ohjelmistoja käyttäviä palveluita on mahdollista muuttaa toimintavarmemmiksi ja suoritustehokkaammiksi. Uudet kehittymismahdollisuudet esitellään vertaamalla nykyisiä ratkaisuja jaetun Unix-ympäristön mahdollistamiin tehokkaampiin toimintamalleihin. Opinnäytetyössä esiteltyt ratkaisut mahdollistavat laadun parantamisen palveluiden toimintavarmuuden nostaessa palveluita käyttävien asiakkaiden tyytyväisyyttä. Ratkaisut oletettavasti pienentävät myös riskien ottamisen tarvetta kohdeympäristössä ja vähentävät asiantuntijoiden työn määrää.

---

Asiasanat: suurtietokone, z/OS, Unix, tiedostojärjestelmä, hajautetut järjestelmät, järjestelmän hallinta

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree programme in Information System Competence, Master's Degree

MALINEN, VELI-MATTI:  
Implementation of Distributed Unix File Systems in z/OS Operation System

Master's thesis 70 pages, appendices 5 pages  
June 2013

---

The Social Insurance Institution of Finland (Kela) made a decision to carry out plans for the distributed Unix file system in their mainframe computer environment in 2010. The problem with the current mainframe environment is segregation of the Unix-systems, which means that the Unix file systems are available only in the system where they are mounted. This restriction makes the duplication of systems in the mainframe cumbersome and it becomes difficult to ensure the performance of the services. It becomes more arduous to verify either the proper functionality of the software, or the integrity of the data, which in turn might affect the services critical to Kela.

The purpose of this study was to describe the distributed Unix file system implemented in the target environment, as well as to introduce the development possibilities the change offers. Due to the uniqueness of the environment setting, a case study approach was chosen.

A number of development propositions were made that would make running services in the mainframe computer both more reliable and efficient. Comparison between the new and the current solutions in Kela showed distributed Unix file systems able to produce a more efficient business model for IT-systems. The solutions presented in the thesis can be expected to improve the quality of service, and therefore raise the level of customer satisfaction. Moreover, the new solutions should reduce the necessity of risk-taking in the mainframe environment and reduce the amount of work needed from IT specialists.

---

Key words: mainframe, z/OS, Unix, file system, distributed systems, system management

## SISÄLLYS

1 JOHDANTO .....	7
2 TAUSTA .....	8
2.1 Suurtietokoneet .....	8
2.2 Kelan IT-osasto .....	9
2.3 Kelan laiteympäristöt .....	11
3 IT-PALVELUIDEN JATKUVUUDEN VARMISTAMINEN .....	12
3.1 IT-palveluiden riskien hallinta .....	12
3.2 Tieto-omaisuus ja sen tunnistaminen .....	15
3.3 Tiedon saatavuuden vaatimuksia valtionhallinnossa .....	16
3.4 Kelan IT-palvelut ja niiden varmistaminen .....	18
4 KESKEISTEN KÄYTTÖJÄRJESTELMIEN MÄÄRITTELY .....	20
4.1 Käyttöjärjestelmä Unix .....	22
4.1.1 Unixin tiedostojärjestelmä .....	25
4.2 Käyttöjärjestelmä z/OS .....	29
4.2.1 z/OS tiedostojärjestelmät .....	33
4.2.2 z/OS käyttöliittymät .....	38
4.2.3 Eräajojen ja töiden hallinta z/OS-käyttöjärjestelmässä .....	42
5 KELAN z/OS-JÄRJESTELMÄYMPÄRISTÖ .....	45
5.1 Jokainen Unix on saari .....	47
5.2 Jaetun USS-ympäristön mahdollistamat hyödyt ja haasteet .....	49
6 JAETTUIJEN UNIX-TIEDOSTORESURSSIEN KÄYTTÖÖNOTTAMINEN .....	52
6.1 Projektin uuden käyttöjärjestelmän käyttöönotosta .....	52
6.2 Unix-tiedostoarkkitehtuurin suunnittelu ja määrittäminen .....	53
6.3 Käyttöönottoa edeltävät muutokset .....	56
6.4 Käyttöönoton toteuttaminen .....	57
6.5 Tiedottaminen, dokumentointi ja mittaaminen .....	58
7 TULOSSIEN TARKASTELU .....	61
7.1 Jaettujen resurssien käyttäminen järjestelmien välillä .....	61
7.2 Uudet kehittämismahdollisuudet .....	62
7.3 Tavoitteiden tarkistus .....	62
8 YHTEENVETO .....	63
LÄHTEET .....	64
Kirjalliset lähteet .....	64
Verkkolähteet .....	64
LIITTEET .....	66

## LYHENTEET JA TERMIT

3270-pääteohjelma	Ohjelma joka emuloi IBM 3270-tietokonepäätteiden toimintaa. 3270-pääteohjelmalla otetaan yleensä yhteys merkki-pohjaiseen kuvaruutukäyttöliittymään, kuten TSO:hon.
CICS	Customer Information Control System. CICS on IBM:n suurtietokoneen järjestelmä, joka tarjoaa mahdollisuuden koodata oman tapahtumankäsittely-ympäristön.
Client/Server	Tietojenkäsittelyn tapa, jossa palvelun tehtäviä jaetaan palvelua pyytävien ohjelmien (client) ja palvelua tarjoavien ohjelmien (server) välille. Toimii yleensä hajautetussa ympäristössä, jossa pyytävät ja tarjoavat ohjelmat sijaitsevat eri palvelimilla.
Eräajo	Ohjelmaryhmien ajastettu peräkkäinen suorittaminen ilman manuaalista operointia. Eräajo voi sisältää ohjelmia muutamasta useisiin tuhansiin.
Hajautetut järjestelmät	Tietokonejärjestelmä, tietoliikennejärjestelmä, tai ohjelmisto, jonka voi katsoa koostuvan itsenäisistä, keskenään kommunikativista osista. Käyttäjälleen hajautettu järjestelmä näyttää yhdeltä tietokoneelta, tai käyttöliittymältä.
HFS	Hierarchical File System. USS:n tiedostojärjestelmän tiedostoresurssityyppi, joka pitää kiinnittää tyhjään hakemistoon. Ei pidä sekoittaa samannimiseen Applen kehittämään tiedostojärjestelmään.
IT	Informaatioteknologia. Tietokoneiden ja tietoliikenteen avulla tapahtuva tiedonhallinta.
Jaetut tiedostoresurssit	Tietojärjestelmä jakaa omistamansa tiedostoresurssin muiden tietojärjestelmien käyttöön.
Käyttöjärjestelmä	Tietokoneen käyttämiseen vaadittava ohjelmisto, joka määrittelee tietokoneen resurssit käyttöä varten ja jonka päällä muut ohjelmat toimivat.
LPAR	Logical partition. IBM:n termi suurtietokoneen virtuaaliselle järjestelmälle, joka ei ole riippuvainen toisista järjestelmistä.

MVS-tiedostojärjestelmä	z/OS käyttöjärjestelmään edeltäjästään MVS käyttöjärjestelmästä periytynyt tiedostojärjestelmä.
Parallel Sysplex	IBM:n termi suurtietokoneessa, tai suurtietokoneissa sijaitsevista z/OS-pohjaisista järjestelmistä, jotka toimivat rinnakkain mahdollistaen tehokkaan tietojenkäsittelyn suorittamisen ja etenkin korkean palvelutason.
Shell	Käyttöjärjestelmän komentotulkki, joka tulkitsee käyttäjän komennot laitteiston ymmärtämään muotoon.
Suurtietokone	Erittäin suorituskykyinen palvelintietokone. Nykyiset suurtietokoneet ovat yleensä IBM:n zSeries-sarjan palvelintietokoneita.
Suurkone	Kts. Suurtietokone
Tiedostojärjestelmä	Käyttöjärjestelmän osa, jonka avulla tietoa voidaan tallentaa massamuistiin.
TSO	Time Sharing Option. Tekstipohjainen kuvaruutukäyttöliittymä suurtietokoneen ohjelmistojen hallintaan.
Unix	Käyttöjärjestelmä, jonka pohjalta on kehitetty lukuisia avoimen ja suljetun lähdekoodin käyttöjärjestelmiä
USS	Unix System Services. z/OS käyttöjärjestelmään sisällytetty ohjelmistokokonaisuus. Sertifioitu Unix-pohjainen käyttöjärjestelmä, joka toimii osana z/OS käyttöjärjestelmää.
Virtuaalinen järjestelmä	Tietokoneen laiteresursseista määritellään osa omaksi kokonaisuudeksi, jossa on mahdollista ajaa omaa käyttöjärjestelmää.
Virtualisointi	Tekniikka jolla jaetaan yksi fyysinen resurssi useaksi loogiseksi resurssiksi tai päinvastoin. Luodun virtuaalisen näkymän takana olevat laitteet tai järjestelmät eivät näy ulospäin.
VSAM	Virtual Storage Access Method. z/OS käyttöjärjestelmän tiedonsaanti- ja tallennusmenetelmä, joka on tarkoitettu ensisijaisesti sovellusten käyttöön.
z/OS	IBM:n käyttöjärjestelmä zSeries-sarjan suurtietokoneille.
zFS	zSeries Files System. USS:n tiedostojärjestelmän tiedostoresurssityyppi, jonka on tarkoitus korvata HFS tyhjiin hakemistoon kiinnitettävänä tiedostojärjestelmänä. zFS käyttää hyväkseen tehokkaampaa VSAM tiedonsaantimenetelmää.

## 1 JOHDANTO

Tiedostoresurssien jakaminen, hajautetut tietojärjestelmät ja järjestelmien virtualisointi ovat jo vuosia olleet arkipäivää yritysten IT-ympäristöjen hallinnassa. Hajauttaminen ja resurssien saatavuuden varmistaminen kuuluvat olennaisena osana strategiseen IT-riskien hallintaan. Huoltotoimenpiteistä tai odottamattomista ongelmista johtuvat käyttökatkokset IT-palveluissa pyritään minimoimaan. Näillä toimenpiteillä yritykset pyrkivät yleisesti tehostamaan tietojärjestelmiensä toimintaa sekä tätä kautta parantamaan palvelujensa vikasietoisuutta, luotettavuutta ja laatua.

Tässä opinnäytetyössä käsitellään case-tutkimustyyppisesti Unix-tiedostoresurssien jakamista, tutkimuskohteen tilaajan laite- ja järjestelmäympäristöön parhaiten soveltuvalla tavalla. Opinnäytetyön tavoitteena on luoda kohdeympäristöön toimintavarmempi ja dynaamisempi tietojärjestelmä, jossa palveluiden kahdentaminen on ympäristön puolesta mahdollista. Tarkoituksena opinnäytetyössä on kuvata Unix-tiedostoresurssien jakaminen tilaajan järjestelmäympäristössä sekä esittää konkreettisia parannusehdotuksia joita muutos kohdeympäristöön mahdollistaa. Tätä tarkoitusta varten pyrin kuvaamaan kohdeympäristön ja sen kriittiset järjestelmät tarkkuudella, joka mahdollista muutoksen vaikutuksen ymmärtämisen.

## 2 TAUSTA

Opinnäytetyön tilaaja ja tutkimusympäristön omistaja on Kansaneläkelaitos eli tuttavallisemmin Kela. Kelan tehtävä on huolehtia Suomen lakien mukaisien etuuksien käsittelystä sosiaaliturvan piiriin kuuluville henkilöille. Kelan etuuskäsittelyyn liittyvien IT-palveluiden ydintoiminnot tapahtuvat suurtietokoneympäristössä, verkkopalveluiden sekä asiakasrajapintojen sijaitessa pienemmillä Intel-pohjaisilla Windows- tai Linux-palvelimilla. Oma asiantuntemukseni sijaitsee pääosin suurtietokonejärjestelmissä, joiden parissa olen työskennellyt asiantuntijana vuodesta 2001 alkaen. Erityispiirteensä tähän opinnäytetyöhön antaa tietojärjestelmälusta jolla Unix-tiedostoresurssit on tarkoitus jakaa. Suurtietokoneet ovat hyvin spesifinen ympäristö, joka eroaa paljon muista, nykyisin paljon yleisimmistä pienpalvelinympäristöistä.

### 2.1 Suurtietokoneet

Ensimmäiset suurtietokoneet kehitettiin 1940-luvulla toisen maailmansodan tarpeisiin. 50-luvulla suurtietokoneita ryhdyttiin valmistamaan kaupallisiin tarkoituksiin ja vuosikymmenen lopulla massatuotantona suuryritykset alkoivat valmistaa suuryritykset IBM valtasi markkinat ja on pitänyt niitä hallussaan siitä asti. Suuryritykset ovat aina olleet suurien yritysten suosiossa. Etenkin rahoitus- ja vakuutuslaitokset, joilla on suuret kaupalliset tietokannat, suosivat nykyäänkin suuryrityksiä IT-palveluidensa laitealustana.

Pitkään markkinoilla toimineet suuryritykset ovat ehtineet rakentaa tietojärjestelmiään sekä kehittää automatisoituja tapahtumankäsittelyrutiinejaan jo useiden vuosikymmenien ajan. Kun tietotekniikka on rakennettu vuosikymmenten saatossa sellaiseksi, että se tukee hyvin nimenomaan organisaation ydintoimintojen suorittamista, on kynnys sen vaihtamiseen korkealla. Nämä ajan saatossa huolella rakennetut suuryritykset koetaan myös yleisesti hyvin vakaiksi ja tietoturvallisiksi ratkaisuksiksi, joissa on panostettu jatkuvuuteen sekä jatkuvakäyttöisyyden varmistamiseen.

Suurimmat syyt, siihen miksi niin monet yritykset kuitenkin pyrkivät korvaamaan suuryritykset muilla palvelimilla, liittyvät kustannuksiin. Pelkät IBM:n Z-sarjan suuryritykset maksavat kahdennettuina vuositasolla useita miljoonia euroja, koneiden tehosta riippuen. Keskitetysti, kahdennettujen Z-sarjan suuryritykset



ohjelmisto-lisenssien vuosikustannukset taas voivat helposti nousta jopa useiden kymmenien miljoonien eurojen arvoiseksi. Lisenssihinnoissa on nähtävissä miten IBM kykenee edelleen käyttämään hyväkseen menneiden vuosikymmenten monopoliasemaansa, jolloin IBM-ratkaisut olivat ainoat mahdolliset tietoarkkitehtuuriset ratkaisut. Pitkän linjan organisaatiot usein maksavat vastentahtoisesti näitä massiivisia lisenssimaksuja, koska vuosikymmenten aikana rakennetun toimivan järjestelmän siirto on myös äärimmäisen kallista, aikaa vievää ja ennen kaikkea siirtämiseen liittyy aina suuria riskejä.

Suurtietokoneympäristöt ovat myös olleet muita tietojärjestelmäalustoja hitaampia muuttumaan. Järjestelmäympäristö, jossa halutaan varmistaa vuosikymmenten aikana rakennettujen ohjelmistojen toiminta ja tähdätä pitkäjänteiseen sekä stabiiliin kehittämiseen, ei kehity harppauksin. Tästä johtuen monet uudet ominaisuudet, jotka tulevat Windows- tai Linux-palvelimille, saapuvat usein vasta vuosia myöhemmin suurtietokoneympäristöön.

Viimeisimpien vuosikymmenten kuluessa suurtietokoneet ovat yrityskauppojen ja ulkoistamisen kautta keskittyneet yhä harvempien toimijoiden konesaleihin. Harvinaistuminen taas johtaa siihen, ettei osaavaa työvoimaa suurtietokoneympäristöihin löydy. Osaajat ovat pääsääntöisesti ikääntyneitä, uransa ehtopuolella olevia henkilöitä ja esimerkiksi Suomesta ei löydy koulua tai oppilaitosta, jossa opetettaisiin mitään suurtietokonejärjestelmistä.

## **2.2 Kelan IT-osasto**

Kelan IT-osasto perustettiin vuonna 1968 vastaamaan uuden tietotekniikan haasteisiin. Nykyään valtaosa Kelan IT-osaston noin 500 työntekijästä työskentelee Helsingissä ja Jyväskylässä. Kelan tietojenkäsittelyn historia esitellään pähkinänkuoressa kuviossa 1.



Kuvio 1. Tietojenkäsittelyn historiaa Kelassa (Kelan IT-toiminnan haasteita 2011).

IT-osaston vastuulla on Kelan tietojärjestelmien tuottaminen, johon sisältyvät sosiaaliturvan etuustietojärjestelmät, sähköiset asiointipalvelut kansalaisille sekä muut tietotekniikkapalvelut Kelalle. Lisäksi IT-osasto vastaa Kelan IT-infrastruktuurista sekä terveydenhuollolle, apteekkeille ja kansalaisille tarjottavista kansallisen terveysarkiston palveluista. Esimerkkejä Kelan IT-palveluiden määristä on esitelty kuviossa 2.

Kela Tunnuslukuja Kelan IT-palveluista	
• Etuuskäsittelyn ohjelmia	9 000
• Päätejärjestelmässä sovellustapahtumia/päivä	8 000 000
• Etuusjärjestelmien vastausaika noin	0,05 s
• Tunnistuksia sähköisiin palveluihin/v	6 500 000
• Kirjeitä asiakkaille vuodessa	18 000 000
• Kaksi IBM-suurkonetta	
• Noin 550 Intel-palvelinta, joista 160 virtualisoitu	

Kuvio 2. Kelan IT-palvelujen tunnuslukuja (Kelan IT-toiminnan haasteita 2011).

Nykyisin kaikki Kelaan tulevat etuushakemukset liiteasiakirjoihin tallennetaan Sähköiseen asianhallintajärjestelmään (SAHA), joka mahdollistaa etuuskäsittelyn sellaisessa toimistossa jossa on juuri sillä hetkellä kapasiteettia hoitaa asia. Tai sitten etuushakemus voidaan järjestelmän kautta ohjata ratkaistavaksi eniten etuuden asiantuntemusta omaavalle yksikölle.

### **2.3 Kelan laiteympäristöt**

Kelalla on pitkä historia suurtietokoneella pyörivistä IT-palveluista. Ensimmäinen tietokone, IBM 650, otettiin käyttöön vuonna 1960. Nykyään vanhimmat tuotantokäytössä edelleen toimivat eräajot on ohjelmoitu 70-luvulla. Kaikki etuuskäsittely tapahtuu edelleen suurkanneen CICS-järjestelmän avulla.

Viimeisen kymmenen vuoden aikana useat Kelan palvelut ovat siirtyneet Internetiin. Tästä johtuen pienpalvelinten lukumäärä on kasvanut Kelassa huomattavasti. Ne ovat olleet looginen valinta laitealustaksi uusille verkkopalveluille, niiden hankintahinnan sekä ohjelmistolisenssien hintojen ollessa merkittävästi suurkanneiden vastaavia edullisempia. Myös laitealustoja hallinnoivia asiantuntijoita on helppo rekrytoida, ainakin verrattaessa suurkanneeseen.

Monet uudet verkossa toimivat asiointipalvelut ohjautuvat kuitenkin Intel-pohjaisten Windows- ja Linux-palvelinten kautta edelleen suurkanneelle hakemaan asiakastietoja tietokannoista, tai suorittamaan kyselyitä etuuskäsittelyjärjestelmistä. Tietoarkkitehtuurin ollessa rakennettu tällä tavalla, Kela on sidottu käyttämään suurkannejärjestelmiin pohjautuvia tietojärjestelmiä vuosiksi eteenpäin. Lisätietoa Kelan palvelinratkaisusta on luettavissa Kauppalehden verkkoartikkelissa, *Kelassa hyrräävät tehokkaat koneet*, <http://www.kauppalehti.fi/5/i/talous/uutiset/avoinarkisto/index.jsp?xid=3070667&date=2008/11/25>.

### 3 IT-PALVELUIDEN JATKUVUUDEN VARMISTAMINEN

Monien organisaatioiden IT-palvelut pitävät pystyssä niiden tärkeimpiä liiketoimintaprosesseja. Liiketoiminta pyörii usein IT-ratkaisuiden varassa ja näiden häiriöt sekä katkokset voivat johtaa mittaviin taloudellisiin tappioihin, tai joskus jopa ihmishenkien vaarantumiseen. Lukuisten yritysten IT-osastot eivät ole omaksuneet IT-palveluiden tarjoamiseen pohjautuvaa mallia, jossa palvelut olisivat keskiössä ja tekniset ratkaisut sekä sovellukset mahdollistavat nämä palvelut. Liian usein keskitytään liikaa tekniikkaan sekä infrastruktuuriin, tietämättä tarkkaan mitä palveluita kaatuneen järjestelmän takana on ollut. (Jordan & Silcock 2005, 15-16.)

Suorituskykyyn vaikuttaa oleellisesti käytettävyys ja luotettavuus. Mikäli IT-palveluiden voidaan katsoa tarjoavan vähintäänkin tyydyttävällä tasolla palvelua, täytyy taustalla olevan tietojärjestelmän myös toimia. Mikäli palvelu ei toimi oikein, jokin sen taustalla olevista järjestelmistä toimii virheellisesti, tai ei toimi ollenkaan. Käytettävyys on järjestelmän kyvykkyyden mittari, jolla mitataan pystytäänkö haluttu palvelu tuottamaan oikein aina kun sitä tarvitaan. Luotettavuudella taas mitataan järjestelmän kykyä toimittaa palvelua oikeellisesti, eli kykyä havaita mahdolliset virheet ja korjata ne ennen vahingon tapahtumista. (Crichlow 2000, 82)

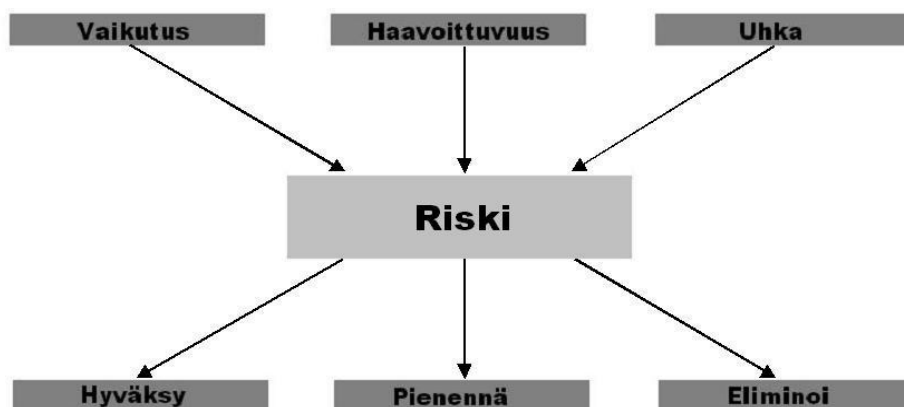
#### 3.1 IT-palveluiden riskien hallinta

Hyvä suunnittelu on kaikkein tärkein osuus IT-palveluiden riskien minimoimiseen liittyvässä työssä. Pahimmilta katastrofeilta onnistuvat välttymään ne organisaatiot, jotka laativat suunnitelmia kriisitilanteiden varalle ja ovat varustautuneet kohtaamaan ongelmia, jotka eivät ole vielä aikaisemmin heidän kohdallaan realisoituneet. Oleelliset palvelut täytyy kyetä priorisoimaan ja niille tulee määritellä hyväksyttävissä oleva käyttökatkosaika, mikäli sellainen ylipäättään palvelun kohdalla on mahdollinen. Hyvänä lähtökohtana tällaiselle kehittämistoiminnalle on organisaation palvelukuvauskartta, johon sen tarjoamat palvelut on kuvattu ympäristöineen ja liittymineen. IT-henkilökunta on tärkeää sitouttaa näihin suunnitelmiin ja pyrkiä iskostamaan heille palveluorientoitunutta ajattelumallia, teknisen ajatusmallin sijaan. Tekniikka ja järjestelmät ovat palveluita varten, eikä toisin päin. Saman järjestelmäkokonaisuuden alla saattaa sijaita useita erilaisen priorisoinnin omaavia palveluita jotka käyttöhenkilökunnan täytyy kyetä asetta-

maan tärkeysjärjestykseen. Esimerkiksi häiriön aikana käyttöhenkilökunnan täytyy kyetä päättämään mitkä palvelut voidaan keskeyttää, jotta resurssit riittävät kriittisimpien palveluiden tarjoamiseen. (Jordan & Silcock 2005, 134-135.)

IT-palveluiden toiminnan turvaamiseen tähtäävät toimet ovat ikään kuin vakuutuksen ottaminen. Niillä pyritään pienentämään mahdollisuus katastrofiin, mikäli jonkin odottamaton tilanne iskee. Aivan kuten vakuutuksen ottaminen, myös IT-palveluiden turvaaminen maksaa ja mitä varmempaa suojaa haluaa, sitä kalliimmaksi se käy. Yksiselitteistä kaavaa siihen, kuinka paljon resursseja varmistukseen ja luotettavuuden parantamiseen kannattaisi käyttää, ei ole. Panostuksen tulisi kuitenkin aina olla suhteessa arvioituun riskiin. Resursseja tulisi aina kohdistaa ensisijaisesti järjestelmien ja palveluiden kriittisimpiin osiin. (Jaakkohuhta 2003, 7-8.)

Palveluiden haavoittuvuuteen liittyvät riskit olisi tärkeää tunnistaa, yksilöidä, arvioida ja hallita. Riskien tunnistamisessa pyritään löytämään ne kaikkein yleisimmät uhkakuvat joiden varalle tehdään suunnitelmat, sillä kaikkien riskien suunnitelmallinen torjunta on käytännössä mahdotonta, osa täytyy vain hyväksyä hallittuna riskinä. (Jordan & Silcock 2005, 135-136.) Kuviossa 3. esimerkki riskien analysoinnista ja niiden aiheuttamista toimintasuunnitelmista.



Kuvio 3. Riskien analysointi ja hallinta (Jaakkohuhta 2003, 10).

Jordan & Silcock (2005, 60) ovat teoksessaan luokitelleet IT-riskit seitsemään erilaiseen kategoriaan:

1. IT-projektien epäonnistuminen
2. IT-palveluiden jatkuvuuden epäonnistuminen
3. Tieto-omaisuuden katoaminen
4. Palveluiden tarjoajista ja IT-toimittajista johtuvat epäonnistumiset
5. Sovelluksista johtuvat ongelmat
6. Infrastruktuurista johtuvat ongelmat
7. Strategiset riskit ja tulevaisuuden uhat

Tässä opinnäytetyössä näkökulma on kategorioiden IT-palveluiden jatkuvuuden epäonnistumisen ja tieto-omaisuuden katoamisen ehkäisyyn pyrkivissä keinoissa.

Yleensä sopivia toimenpiteitä tunnistettujen riskien vähentämiseksi ovat esimerkiksi resurssien kahdennus, virheidensietokyvyn kasvattaminen ja varajärjestelmien perustaminen. Tavoitteena on yleensä palveluiden kestävyvyn tehostaminen, siten että ne sietäisivät tietyn määrän katkoksia, kyeten silti tarjoamaan tavoiteltua suorituskykyä. Kestävyyden suunnittelussa tulee varautua myös kriisitilanteisiin. Ongelmatilanteissa kriittisille palveluille tulisi aina löytyä kapasiteettia, vähemmän kriittisten joutuessa tyytymään vähempään. Toinen yleisesti tärkeä tavoite on kehittää organisaation kykyä palautua ongelmista. Kriisin iskettyä tulisi siis löytyä suunnitelma jonka mukaan työntekijät osaavat toimia ja riittävät laitteistoresurssit joilla taataan vikatilanteissakin tärkeimpien palveluiden toiminta. Näillä ratkaisuilla palvelut voidaan yleensä palauttaa nopeasti normaalitilanteisiin, siten etteivät palveluita käyttävät tahot välttämättä edes huomaa mitään tapahtuneen. (Jordan & Silcock 2005, 16 ja 136.)

Palveluiden käyttöön tulee siis tarkoituksellisesti varata ylimääräisiä resursseja sekä pyrkiä käyttämään mahdollisimman vikasietoisia laitekokonaisuuksia. Infrastruktuuria kehittämällä, eli hankkimalla varajärjestelmiä ja kahdentamalla tekniikkaa voidaan riskejä pienentää. Lisäksi henkilökuntaa voidaan kouluttaa kohtaamaan kriisejä ja ongelmatilanteita erilaisten harjoitusten ja testausten avulla. Uhkakuviin ja riskeihin valmistautuessa on muistettava että kustannukset nousevat aina sitä suuremmaksi, mitä parempaa kestävykyä palveluille suunnitellaan. Toisaalta organisaatiolle tulee aina halvemaksi sisällyttää alusta alkaen omiin prosesseihinsa määritelmät kestävyvystä ja

palautumisstrategiasta, kuin lisätä ne vasta jälkikäteen. (Jordan & Silcock 2005, 136 ja 148.)

### 3.2 Tieto-omaisuus ja sen tunnistaminen

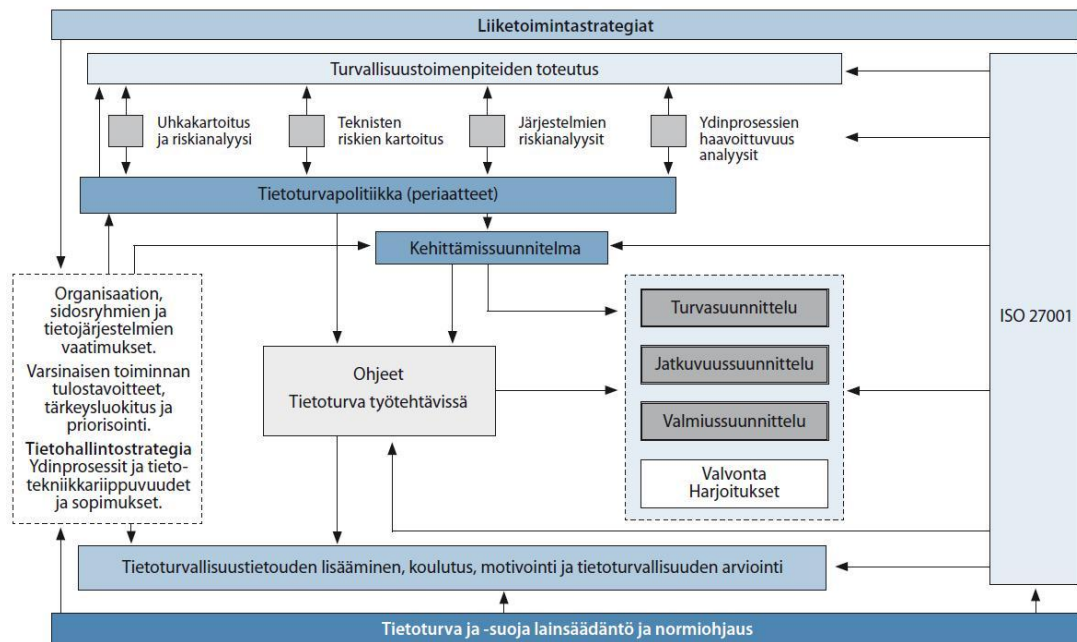
Jokaisella organisaatiolla on hallussaan informaatiota jota se käyttää, tai jolla se mahdollistaa muiden etujen tuottamisen. Tätä näkymätöntä pääomaa kutsutaan tieto-omaisuudeksi. Tieto-omaisuuden saatavuus oikeille henkilöille oikeaan aikaan on organisaation prosessien toimivuuden kannalta ensiarvoisen tärkeää. Tämä tarkoittaa sitä että organisaatioiden tulee suunnitelmallisesti tunnistaa, hallita ja tallentaa tieto-omaisuutensa (Jordan & Silcock 2005, 157-158).

Tieto-omaisuus hajaantuu usein moniin eri paikkoihin organisaation sisällä; palvelimelle, tietokantaan, työasemaan, verkkolevylle, nauhavarmistukselle, usb-muistille, tai vaikka tulostettuina dokumentteina työntekijän pöydänkulmalle. Kaiken tämän tiedon hallintaan on tarpeen olla kattavat menettelyohjeet, joiden avulla pelisäännöt tehdään selväksi organisaation laajuisesti. Ohjeiden taustalla vaikuttaa yleensä suuresti paikallinen tai kansainvälinen lainsäädäntö. Suomessa valtionhallinnon alaisissa organisaatioissa tieto-omaisuudenhallinnan ohjeistuksen määrittelee Valtionhallinnon tietoturvallisuuden johtoryhmä (VAHTI).

Tärkeän tieto-omaisuuden tulisi olla yhdenaikaisesti käytettävissä siihen oikeutetuilla ihmisillä. Tästä syystä aikojen saatossa on kehitetty monenlaisia tiedon jakamisen mahdollistavia tekniikoita, joilla mahdollistetaan käyttäjien tunnistaminen, oikeuksien antaminen, ajoitus, pääsyn koordinointi, tiedon synkronointi sekä ristiriitatilanteiden ratkaiseminen. Jakaminen, joka on hajautettujen järjestelmien avainominaisuus, tuo käyttäjät ja tieto-omaisuuden yhteen ja mahdollistaa tehokkaamman sekä joustavamman yhdessä työskentelyn. (Crichlow 2000, 14 ja 79.)

Tiedonhallinnan alalla on julkaistu paljon standardeja ja suosituksia. Suomessa kenties tunnetuimmat tällaiset suositukset ovat Julkisen hallinnon tietohallinnon (JUHTA) julkaisemat JHS-suositukset. VAHTI taas käyttää Tietoturvallisuuden johtamisen ja hallinnan ohjeessa avuksi kansainvälistä ISO 27001 standardia (VAHTI 3/2007, 38-39).

Kuviossa 4. VAHTIn ohjeistuksen mukainen tietoturvallisuuden hallintajärjestelmän malli.



Kuvio 4. Tietoturvalisuuden hallintajärjestelmän malli (VAHTI 3/2007, 41).

### 3.3 Tiedon saatavuuden vaatimuksia valtionhallinnossa

Valtionhallinnon alaisilta organisaatioilta vaaditaan tietoturvalisuuteen liittyvissä ohjeistuksissa erityistä huomiota tiedon oikea-aikaiselle saatavuudelle ja käytettävyydelle. Kela ei ole valtionhallinnon alainen organisaatio, mutta on monessa mielessä hyvin verrannollinen niihin. Valtionhallinnon tietoturvalisuuden ohjeet voisivat ihan hyvin toimia ohjesääntöinä myös Kelassa.

Tiedon, tietojärjestelmän tai palvelun on oltava sen käytettävyyysvaatimuksen määrittelemässä ajassa ja ajankohdassa siihen oikeutetun henkilön, tai prosessin saatavilla ja käytettävissä. Tietojen saatavuus tulee rajoittaa siten että se on vain siihen oikeutettujen käytettävissä, riippumatta tiedon muodosta, tai muista ympäristön muuttujista. Tiedon muuttumattomuudesta tulee voida olla varmuus ja tiedon mahdolliset muutosvaiheet tulee kirjautua, jotta ne voidaan kiistattomasti tunnistaa. Tiedon



käytettävyyksivaatimus heijastuu usein hyvästä tiedonhallintatavan toteuttamistarpeesta. (VAHTI 2/2010, 32)

Nykyään kansalaiset ovat tottuneet vaatimaan reaaliaikaista palvelua internetpalveluiden kautta. Tämä on asettanut paineita myös valtionhallinnon alaisiin organisaatioihin, jotka pyrkivät viemään asiakaspalveluitaan yhä enemmän verkkoon. Moniin tärkeisiin palveluihin on määritelty korkea käytettävyystvoite, joka saattaa lähennellä 100 prosenttia, jättäen vaikkapa vain yhden tunnin huoltokatkon viikossa. Valtionhallinnon verkkopalveluissa käsitellään usein asiakkaan henkilökohtaisia tietoja, tästä johtuen käyttäjän tunnistaminen on tärkeässä roolissa. Yleisin tunnistautumistapa tällä hetkellä on tunnistautuminen verkkopankkitunnuksia käyttäen.

Lukuisat toimintaprosessit vaativat tiedolta oikea-aikaista ja eheää saatavuutta. Asiakirjoilla ja tiedolla on yleensä hyvin erilaisia käyttötarpeita. Toimintaprosessin vaatiessa korkeaa käytettävyyttä ja käsitellessä korkean suojaustason asiakirjoja tai tietoja, korostuvat tiedon käsittelyyn liittyvät saatavuuden vaatimukset. Etenkin valvontajärjestelmien tuottama ja käsittelemä data on usein edellä kuvattua tietoa. Tiedon saatavuuteen ja käytettävyyteen liittyvät vaatimukset riippuvat tiedon merkityksestä sitä työssään käyttävälle, tai siihen oikeutetulle. Tiedonkäsittelyn vaatimukset tulisi kyetä ottamaan huomioon jo tietojärjestelmien suunnittelusta alkaen, jolloin olisi helpointa tunnistaa ja määritellä kriittinen tieto sekä varmistaa että sen saatavuus on turvattu ja varmennettu. Tieto jonka välitöntä saatavuutta jokin toiminta vaatii, on kriittistä tietoa. (VAHTI 2/2010, 61-62)

Tiedon saatavuuteen vaikuttavat suuresti käytetty infrastruktuuri, tietoliikenneverkot, varmenteiden käyttäminen, tietojärjestelmän ja työaseman ominaisuudet sekä käyttäjän osaaminen. Prosesseissa käytettävä tieto voi olla julkista, salassa pidettävää tai sitten saatetaan käyttää molempia. Käytettävän tiedon tunnistaminen on ehdottoman tärkeää. Etenkin julkisten asiakirjojen saatavuuden ja eheyden kohdalle asetettujen vaatimusten toteutumiseen tulee kiinnittää erityistä huomiota. Tällä tarkoitetaan sitä että tarvittavat tiedot ovat niitä työtehtävissään tarvitsevien käytettävissä helposti ja työprosessin vaatimalla tehokkuudella. Tarkat määritelmät julkisten asiakirjojen luovuttamisesta niitä pyytävälle on määritelty julkisuuslaissa. (VAHTI 2/2010, 32 ja 62)

### 3.4 Kelan IT-palvelut ja niiden varmistaminen

Kelan IT-järjestelmät voidaan jakaa karkeasti kahteen osaan, etuustietojärjestelmään ja siihen läheisesti liittyviin palveluihin jotka toimivat suurtietokonealustalla sekä kansalaisille tarkoitettuun sähköiseen asiointipalveluun joka toimii Intel-alustalla.

IT-palveluita ajatellessa jako ei kuitenkaan ole yhtä yksinkertainen, sillä kansalaisten asiointipalvelu tarvitsee toimiakseen myös suurtietokoneella toimivan etuustietojärjestelmän, josta asiakkaiden etuuksien tiedot löytyvät. Asiointipalvelu on siis tyypillinen hajautettu järjestelmä, joka käyttäjälleen näyttäytyy yhden käyttöliittymän kautta, käyttäen kuitenkin taustalla useita toisiinsa kytkeytyneitä järjestelmiä (Crichlow, 2000, 1). Ongelmana tällaisessa isojen järjestelmien välisessä integroinnissa ovat vikatilanteet. Vian paikantaminen nimittäin saattaa olla todella haastavaa hajautetun järjestelmän laajuudesta johtuen ja yksittäisten järjestelmien vastuuhenkilöt syyttävät yleensä herkästi sitä toista järjestelmää.

Verkossa toimivat asiointipalvelut ovat oleellisesti muuttaneet Kelan palveluiden käytettävyyksivaatimuksissa määriteltyjä palveluajankohtia. Aikaisemmin, kun etuustietojärjestelmää käyttivät vain Kelan toimistoissa työskentelevät etuuskäsittelijät, olivat palveluajat pääsääntöisesti klo 07.00 – 18.00 ja vain arkipäivisin. Tällöin järjestelmien huoltamiseen liittyvät toimenpiteet oli helppo toteuttaa. Nykyisin asiointipalveluille on määritelty käytettävyydeksi 7 päivää viikossa ja 24 tuntia päivässä, pois lukien sunnuntai klo 02.30 – 04.00 johon on määritelty viikoittainen huoltokatko. Koska asiointipalvelut tarvitsevat taustalle myös etuustietojärjestelmää, ovat myös sen käytettävyyksivaatimukset tiukentuneet samalle tasolle. Tämä on lisännyt oleellisesti järjestelmien huoltotoimenpiteiden suorittamisen haasteellisuutta ja vaatinut erityisesti suunnitelmallista resurssien kahdentamista, järjestelmien virheidensietokyvyn kasvattamista sekä varajärjestelmien perustamista. Kelan asiointipalveluita pidetään yleisesti varsin onnistuneina verkkopalveluina. Konsultointiyritysten järjestämän kyselyn mukaan Kelan verkkoasiointi sijoittui suurien yritysten ja yhteisöjen parhaaseen kymmenykseen (Taloussanomien 2010).

Vuonna 1983 silloinen Kelan ATK-keskus sekä kaikki oleelliset järjestelmät siirrettiin Helsingistä Jyväskylään, osana turvallisuus- ja kriisivalmiussuunnitelmaa (Häggman, 1997, 215). Nykyään Kelan laiteympäristöt on jaettu kahteen, fyysisesti erilliseen palosuojattuun konesaliin. Molemmat konesalit on varustettu VAHTI (1/2002, 23-24)

turvallisuusohjeen mukaisella keskeytymättömällä sähkösaannilla (UPS) sekä varavoimalla. Suurtietokoneita on sijoitettu yksi molempiin konesaleihin. Lisäksi molemmat suurtietokoneet on jaettu useampaan LPAR:iin, eli virtuaaliseen järjestelmään, jotta varmistettaisiin kriittisten palveluiden tarjoaminen useammankin järjestelmän vikaantuessa. Myös Windows- ja Linux-palvelimia on pyritty jakamaan siten että samaa palvelua tarjoavat palvelimet sijaitsivat mahdollisuuksien mukaan eri konesaleissa. Levylaitteistoja on hajasijoitettu tasaisesti molempiin saleihin.

Palvelinten välinen sekä sisään ja ulos suuntautuva tietoliikenne on rakennettu käyttämään kahta toisistaan riippumatonta väylää erillisiin tietoliikennekytkimiin. Kytkinten välillä on yhteys, joka mahdollistaa kaiken liikenteen toisen kautta jommankumman mahdollisesti vikaantuessa. Palvelinten välisen kuorman tasoittamiseen käytetään sekä erillisiä kuormanjakolaitteita, että kuormajakamista varten suunniteltuja ohjelmistoja. Tuotannollisista palveluista otetaan säännönmukaisesti varmuuskopiota, joita säilytetään määriteltävissä olevan ajan joko levylaitteilla, tai kasettivarmituksilla. Varmuuskopioita voidaan palauttaa joko alkuperäiseen sijaintiin tai erikseen määriteltävään paikkaan.

Viimeisiä tuotannollisia palveluita joita ei ole ollut mahdollista kahdentaa järkevällä tavalla on ollut suurtietokoneen Unix-tiedostoresurssit. Siinä missä Windows- ja Linux-palvelimilla sekä suurtietokoneen MVS-tiedostojärjestelmässä on ollut helppo jakaa tiedostoresursseja useiden erillisten järjestelmien kesken, on jakaminen suurtietokoneen Unixin alla ollut käyttöjärjestelmän rajoittuneisuudesta johtuen mahdotonta. Vuonna 2010 julkaistu suurtietokoneen käyttöjärjestelmä z/OS version 1. release 12. sisältää ominaisuuden, jolla on mahdollista jakaa Unix-resursseja suurtietokoneen järjestelmien välillä. Kelan IT-osasto esitti syksyllä 2010 projektia tämän uuden käyttöjärjestelmän asentamisesta suurtietokoneen järjestelmiin. Tämän opinnäytetyön taustalla oleva käytännön kehittämishanke on tuon projektin osakokonaisuus, johon kuuluu jaettujen Unix-tiedostoresurssien muodostaminen ja käyttöönotto suurtietokoneella sijaitsevien järjestelmien käytettäväksi.

#### 4 KESKEISTEN KÄYTTÖJÄRJESTELMIEN MÄÄRITTELY

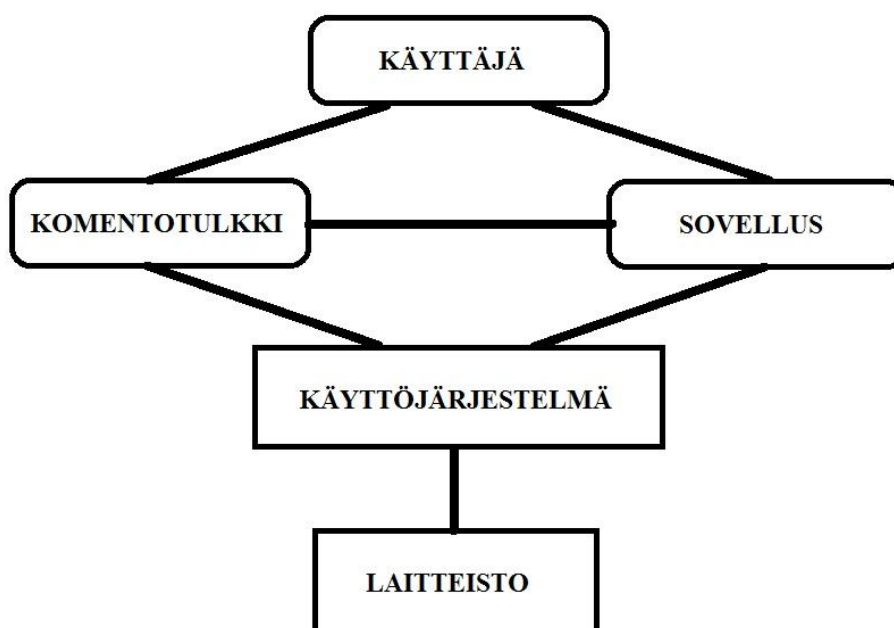
Pelkillä tietokonelaitteistoilla itsessään ei tee mitään. Ne vaativat sopivat järjestelmät valjastamaan laitteiden mukanaan tuoman tehokkuuden haluttuun käyttötarkoitukseen. Käyttöjärjestelmät ovat käytännössä pakollisia ohjelmistoja tietokoneen käyttämistä ajatellen, sillä niillä määritellään laiteresurssit muiden sovellusten käytettäväksi.

Käyttöjärjestelmän tärkein tehtävä on mahdollistaa laitteiden ja sovellusohjelmistojen välinen vuorovaikutus. Käyttöjärjestelmän näkyvin osa on käyttöliittymä, jonka kautta järjestelmää käytetään ja jota usein virheellisesti pidetään itsessään käyttöjärjestelmänä. Yhdellä käyttöjärjestelmällä voi kuitenkin olla monia erilaisia ja erinäköisiä käyttöliittymiä. Tämän tutkimuksen kannalta yksi käyttöjärjestelmän tärkeimmistä osista on tiedostojärjestelmä, jonka tehtäviin kuuluvat tiedostojen talletusrakenteen, hakemistojen ja puskuroinnin hallinta. Muiden ohjelmistojen näkökulmasta käyttöjärjestelmä tarjoaa niiden tarvitsemat oheislaitteiden ohjaukseen, muistinhallintaan ja resurssien sekä oikeuksien hallintaan liittyvät palvelut. Sovellusohjelmoijan kannalta käyttöjärjestelmä mahdollistaa kommunikoinnin järjestelmän kanssa, esimerkiksi ohjelman käynnistämisen järjestelmän muistiavaruuteen. Käyttöjärjestelmiä jotka soveltuvat palvelintietokoneiden sekä työasemien käyttöjärjestelmiksi, kutsutaan yleiskäyttöisiksi käyttöjärjestelmiksi. (Haikala & Järvinen, 2004, 11 ja 183)

Mikäli tietokonejärjestelmää tarkastellaan hierarkkisiin tasoihin jaoteltuna, sijaitsee käyttöjärjestelmä-taso heti laitteisto-tason yläpuolella (Haikala & Järvinen, 2004, 13), kuten kuvioissa 5 ja 6.



Kuvio 5. Tietokonejärjestelmä kuvattuna hierarkkisena koneena (mukaillen Haikala & Järvinen, 2004, 13).



Kuvio 6. Käyttöjärjestelmän periaatekuva (Oksanen 1992, 15).

Haikala & Järvinen (2004, 16 ja 215) jakavat teoksessaan käyttöjärjestelmät viiteen sukupolveen, alkaen sukupolvesta 0 ja päättyen kehittyneimpään verkkokäyttöjärjestelmien sukupolveen 4. Verkkokäyttöjärjestelmällä tarkoitetaan yleensä käyttöjärjestelmää, joka ei toimi paikallisella työasemalla vaan palvelimilla, joihin otetaan tietoverkon kautta yhteyttä. Haikala & Järvinen (2004, 219-221) nimittävät neljännen sukupolven käyttöjärjestelmiä hajautetuiksi käyttöjärjestelmiksi, joiden erityispiirteisiin kuuluvat seuraava ominaisuudet:

- Resurssien jakaminen on toteutettu käyttöjärjestelmän hallitsemien palvelinlaitteistojen kesken.
- Järjestelmän skaalautuvuus, eli järjestelmään voidaan lisätä resursseja joustavasti.
- Järjestelmä on vikasietoinen siten, ettei yksittäisen komponentin vikaantuminen estä järjestelmän käyttämistä ja järjestelmä toimii määrittelynsä mukaisesti kaikissa olosuhteissa.
- Käyttöliittymän räätälöinti yksittäiselle käyttäjälle on helppoa.
- Käyttäjä voi ottaa yhteyden käyttöjärjestelmään tietoverkon läpi, ja hänen tiedostonsa ovat käytettävissä yhteyden muodostuksen jälkeen.

- Ohjelmoinnissa ei tarvitse ottaa erikseen huomioon verkkoarkkitehtuuria, koska verkkoarkkitehtuuri muodostuu suoraan käyttöjärjestelmän avulla.
- Käyttöjärjestelmän ylläpito on helppoa, ei tarvitse päivittää erikseen kaikkien käyttäjien työasemia.
- Käyttöjärjestelmän rakenne on toteutettu hajautetusti, esimerkiksi tiedostojärjestelmä ja prosessien hallinta toimitettiin hajautettuina.

Hajautettujen käyttöjärjestelmien tärkeimpiä ominaisuuksia on tuntumattomuus (transparency). Tuntumattomuudella tarkoitetaan sitä, ettei käyttäjän kannalta ole väliä min-kälaiselta laitteelta hän ottaa yhteyttä käyttöjärjestelmään, vaan järjestelmä toimii yhtenevästi kaikkialta. Käyttöjärjestelmä hoitaa myös resurssit, kuten palvelimet, levylaitteet, tiedostot ja sovellukset käyttäjälle siten ettei hänen tarvitse tietää käyttämiensä resurssien todellista sijaintia. Tämä helpottaa suuresti muutosten tekemistä käyttöjärjestelmässä. (Haikala & Järvinen, 2004, 222)

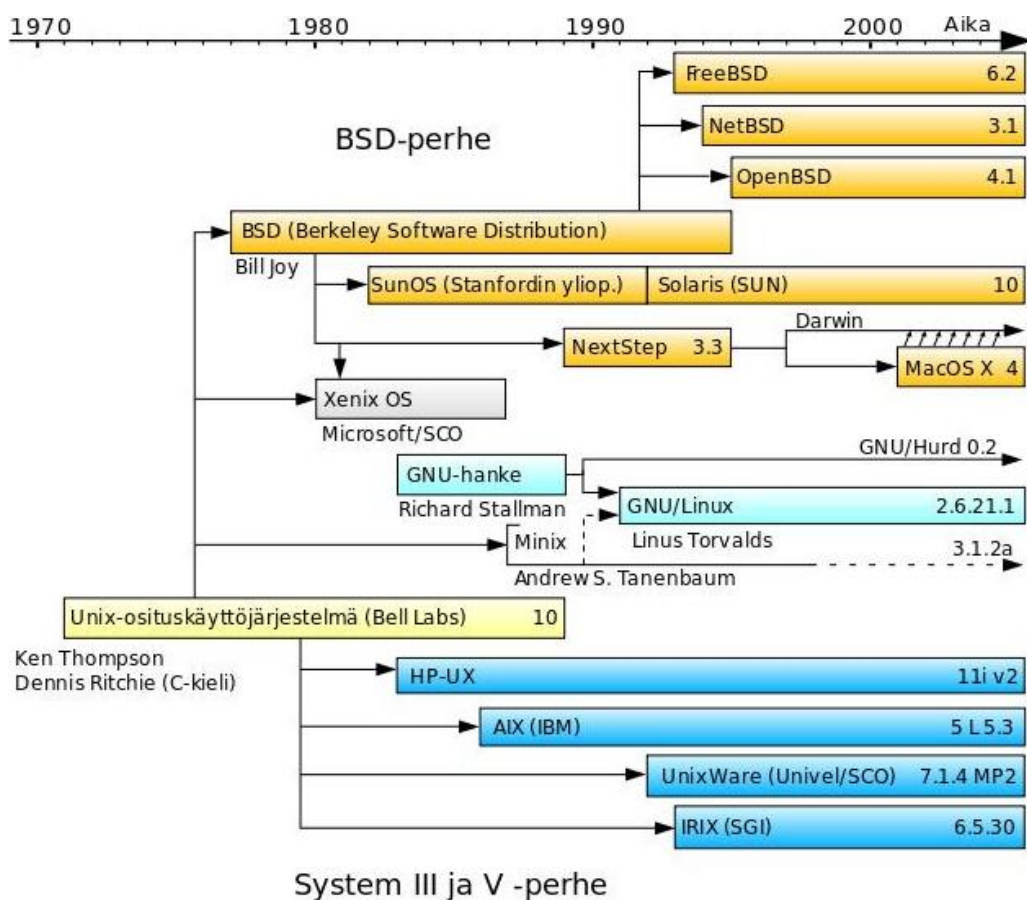
Tähän opinnäytetyöhön sidoksissa olevat käyttöjärjestelmät sisältävät vähintään osia neljännen sukupolven käyttöjärjestelmien erityispiirteistä ja kykenevät pääpiirteissään noudattamaan myös tuntumattomuuden periaatteita.

#### **4.1 Käyttöjärjestelmä Unix**

Unix on yleiskäyttöinen, laitteistoriippumaton käyttöjärjestelmä, josta on saatavissa erilaisia versioita työasemista suurtietokoneisiin. Ensimmäinen Unix ilmestyi 70-luvun alussa. Sen kehittäjät, AT&T:llä (American Telephone and Telegraph) työskennelleet Dennis M. Ritchie ja Ken Thompson, olivat olleet mukana tekemässä MULTICS-käyttöjärjestelmää. Siitä heille oli jäänyt ajatus kehittää moniajokäyttöjärjestelmä, joka olisi yksinkertainen ja kevyt, mutta tehokas ohjelmistokehitysympäristö. Varhaisissa alkuvaiheissa kehitetyt Unixin tiedostorakenne, prosessien toteutus sekä komentorakenne ovat edelleen myös nykyisten versioiden käytössä. Ensimmäisten Unix versioiden yhteydessä kehitettiin käyttöjärjestelmän koodaamista varten uusi korkeamman tason lausekieli, C-kieli, josta tuli hyvin suosittu ohjelmointikieli ja josta löytyy nykyään omat versionsa lähes kaikille laitealustoille. Unix oli ensimmäinen korkeamman tason lausekielellä koodattu käyttöjärjestelmä. C-kielen käyttäminen mahdollisti nopean uudelleen koodaamisen erilaisille laitealustoille. (Oksanen 1992, 9-11.)

AT&T:ltä Unix levisi useammalle yliopistolle, joissa Unixin kehittämistä jatkettiin omaan suuntaansa. Kaliforniassa, Berkleyn yliopistossa kehitettiin Berkley Software Distribution (BSD), josta muodostui myöhemmin yliopistomaailman kehittämän, tutkimuskäyttöön tarkoitetun vapaammin levitettävän Unixin päähaara. Myös AT&T jatkoi Unixin kehittämistä, julkaisten 80-luvun lopulla Unixista System V Release 4 (SVR4), josta muodostui nykyisten kaupallisten Unixien kantaversio. (Oksanen 1992, 11-12.)

Myöhemmin molemmista Unix-haaroista kehitettiin useita enemmän tai vähemmän toisistaan poikkeavia versioita, eikä oikeudenkäynneiltäkään vältytty. Ilmaisen käyttöjärjestelmästä innostuneet tahot halusivat kehittää aidosti vapaan käyttöjärjestelmän, jota ei voitaisi rajoittaa kaupallisten lisenssien taholta. Pitkälti BSD:n pohjalta syntyi GNU-projekti (GNU's Not Unix), joka oli alkusysäys Unixin kaltaisille, tai siitä polveutuville käyttöjärjestelmille, sellaisille kuten Linux (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 3). Unix versioiden haarautumista yksinkertaistettuna esitetään kuviossa 7.



Kuvio 7. Unixien sukupuu ([commons.wikipedia.org/wiki/File:Unixin\\_historia.svg](https://commons.wikipedia.org/wiki/File:Unixin_historia.svg)).

Unix käyttöjärjestelmän määrittelyminen ei ole ongelmaton, kuten eri versioiden haurautumisesta voi päätellä. Tätä ongelmaa helpottamaan luotiin useita virallisia standardeja, joiden avulla oli tarkoitus määrittää yhteensopivuus eri tahoilla kehitettyjen Unixien välillä. Perinteisin ja tunnetuin näistä standardeista on IEEE:n julkaisema Portable Operating System Interface X (POSIX), joka pohjautuu sekä AT&T:n että BSD:n Unix-versioihin. POSIX:lla määritellään Unixin ja sovellusohjelmien rajapinta, joka tarkoittaa ainakin periaatteessa sitä että POSIX-standardia käyttävälle Unixille toteutettu ohjelmisto on käytettävissä myös toisessa POSIX-standardin mukaisessa Unix versiossa. (Oksanen 1992, 355-356.) Suuret laite- sekä järjestelmätoimittajat halusivat myös kehittää omat Unix määritelmänsä. Toimittajista koostunut The Open Group -konsortio, sai vuonna 1994 Novell:ta haltuunsa Unix-tavaramerkin ja alkoi muodostaa virallisia Single UNIX Specification -standardeja (What is UNIX®?). Nykyään sekä POSIX:ia, että Single UNIX Specification standardeja kehittää ja hallinnoi The Open Group -konsortion alla teknisistä asiantuntijoista koostuva Austin Common Standards Revision Group (The Austin Common Standards Revision Group).

Oksasen (1992, 22) mukaan Unixin perusrakenne voidaan jakaa neljään perusosaan:

- Kernel
- Scheduler
- Shell
- Tiedostojärjestelmä

Kernel, eli Unixin ydin ja koneenläheisin osa, muodostuu joukosta aliohjelmia. Sovellusohjelmien ja Kernelin välinen rajapinta koostuu näiden aliohjelmien kutsuista, jotka välittävät sovellusohjelmille niiden tarvitsemat ytimen perustoimenpiteet. Tyypillisiä tällaisia perustoimenpiteitä ovat uuden prosessin käynnistäminen, tiedoston käyttäminen ja muistin varaaminen. (Oksanen 1992, 21) Tärkeimpiä tehtäviä näistä on prosessien muodostaminen ja niiden hallinta, sillä prosessit kutsuvat usein toisia prosesseja ja haurautuvat toimien vuorovaikutuksessa. Koska Kernel on koneen läheisin osa käyttöjärjestelmässä, eivät eri laitealustoille koodatut Kernelit ole toistensa kanssa yhteensopivia. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 4)

Scheduler, eli jakelija, on ohjelma joka mahdollistaa usean henkilön käyttäen järjestelmää yhtä aikaa. Jakelijan tehtävä on jakaa laitteiston resursseja käyttäjien kesken siten, että



käyttäjät saavat vuorollaan suoritusaikaa prosessorilta. Suorituksessa olevan ohjelman sitä osaa, joka sijaitsee keskusmuistissa, nimitetään prosessiksi. Jakelija vuorottelee prosessien kanssa siten, että jokainen saa vuorollaan ajanjakson prosessorilta, jonka jälkeen vuorottelu alkaa alusta vielä suorituksessa olevien prosessien kesken. Mikäli prosesseja on paljon ja kuorma kasvaa, aiheutuu järjestelmässä hidastelua, suorituksessa olevien prosessien joutuessa odottamaan pitkään vuoroaan. Jakelijan tehtävänä on myös hallita muistinkäyttöä siten, että keskusmuisti riittäisi kaikille käyttäjille ja prosesseille. Tällöin odottavat prosessit voidaan tallentaa tilapäisesti levyille (swapping), tai tallentaa ainoastaan osia prosessista (paging), joka on tehokkaampi tapa verrattuna koko prosessin tallentamiseen. (Oksanen 1992, 23-24.)

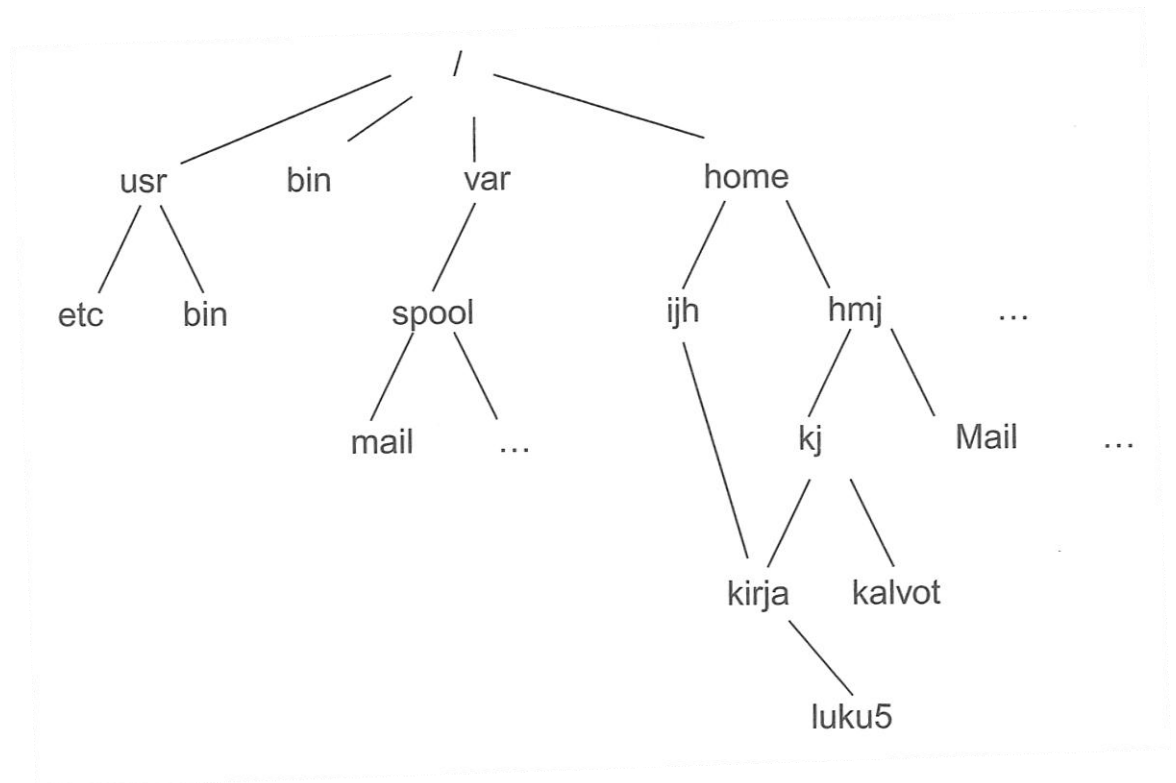
Shell, eli järjestelmän komentotulkki, kääntää käyttäjän kirjoittamalla antamat komennot laitteiston ymmärtämään muotoon (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 5). Shell on ohjelma, joka mahdollistaa järjestelmän käyttämisen, tästä syystä se ladataan jokaiselle käyttäjälle hänen kirjautuessaan järjestelmään. Yksi Unixin vaihtuvuoksista on sen eri versioiden mukana kehitetyt komentotulkit, jotka mahdollistavat tehokkuuden erilaisissa tilanteissa. Yleisimmät komentotulkit ovat: *Bourne shell*, *C shell* ja *Korn shell*. Yhdellä käyttäjällä on mahdollista olla useita komentotulkkeja auki yhtä aikaa. Käyttäjällä on mahdollista vaikuttaa omaan komentotulkkiinsa, muokkaamalla siihen liittyviä muuttujia sekä henkilökohtaisia ympäristömuuttujia. (Oksanen 1992, 26-27.) Nykyään Unixien mukana tulee yleensä myös graafinen käyttöliittymä, eli työpöytäympäristö, joka mahdollistaa Windows-käyttöjärjestelmän tyyllisen hallinnon Unixissa. Yleisimmät komentotulkit, paitsi erillisen lisenssin vaativa Korn shell, löytyvät kuitenkin käytännössä kaikista Unix -järjestelmistä (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 5).

#### **4.1.1 Unixin tiedostojärjestelmä**

Unixin tiedostojärjestelmä rakentuu tiedostoista ja hakemistoista sekä näiden käyttämisen mahdollistavista tiedoista. Unix oli ensimmäisiä käyttöjärjestelmiä, jonka tiedostojen käsittely perustui hierarkkisiin hakemistoihin. Hierarkkisuus helpotti suuresti tiedostojen järjestelyä ja ylhäältä alaspäin olevat tasot selkeyttivät ryhmittelyä. (Oksanen 1992, 26) Nykyään hierarkkinen, eli puumainen, hakemistorakenne on käytännössä standardi kaikissa yleisimmissä käyttöjärjestelmissä.

Lähes kaikkea Unixissa käsitellään tiedostoina, mukaan lukien fyysisiä lisälaitteita, kuten levyasemia ja USB-laitteita. Resurssien ja koko järjestelmän hallitseminen oikeita tiedostoja käsittelemällä, antaa hyvän kuvan siitä, kuinka ehdottoman tärkeä osa Unixia tiedostojärjestelmä on. Lisäksi toteutustapa tekee hallinnasta tavallaan suoraviivaista ja johdonmukaista. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 54, 59)

Unixissa tiedostojärjestelmä alkaa juurihakemistosta, jota yleisesti kuvataan kauttavivalla (/). Juurihakemiston alle sijoittuvat yleensä kaikissa Unix versioissa olevat vakiohakemistot, kuten esimerkiksi `usr`, `bin` ja `var` -hakemistot. Juuresta alkaen hakemistot levittäytyvät ”puumaisesti” haarautuen, yhä alemmille tasoille. Tietyn tiedoston löytämiseksi on siis tiedettävä sen sijainti hakemistorakenteessa sekä tiedoston nimi. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 54). Kuviossa 8. on esimerkki hierarkkisesta hakemistorakenteesta.



Kuvio 8. Tyypillinen Unixin hakemistorakenne (Haikala & Järvinen 2004, 187).

Jokaiseen Unix-käyttäjätunnukseen on määritelty kotihakemisto, josta käyttäjä aloittaa toimintansa järjestelmässä istunnon aloitettuaan. Käyttäjätunnus toimii omistajana kotihakemistolle, ja sen alla käyttäjällä on yleensä oikeus luoda omia hakemistorakenteitaan sekä tiedostoja, tilanvarauksen sallimissa rajoissa. Rajoitetuilla oikeuksilla varustetun

käyttäjän saattaakin olla vain ja ainoastaan mahdollista työskennellä oman kotihakemistonsa alla, muun hakemistorakenteen ollessa hänen käyttäjätunnuksensa oikeuksien ulkopuolelle rajattua. Kotihakemistot sijaitsevat yleensä yhdessä seuraavista kolmesta, juuren alla sijaitsevasta, hakemistosta; /usr, /users, tai /u. Käyttäjän liikkua hakemistorakenteessa, kutsutaan kulloistakin sijaintia hakemistorakenteessa *työhakemistoksi*. (Oksanen 1992, 36-37.)

Huomioon otettavia seikkoja Unixin tiedostojen kanssa on isojen ja pienien kirjainten erottelu tiedostojen nimissä, sekä absoluuttisen – ja suhteellisen polun välinen ero. Toisin kuin monet muut käyttöjärjestelmät, Unix erottelee tiedostojensa nimissä isot ja pienet kirjaimet, eli file1- ja File1-tiedostot voivat sijaita samassa hakemistossa ja voivat sisältää kokonaan eri tiedot. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 56) Absoluuttisella polulla tarkoitetaan hakemistorakenteen kuvaamista juuresta alkaen kohdetiedostoon saakka, esimerkiksi seuraavalla tavalla **/home/ijh/file1**. Suhteellisella polulla taas viitataan siihen hakemistorakenteen kohtaan jossa käyttäjä juuri sillä hetkellä sijaitsee, esimerkiksi käyttäjän ollessa hakemistossa /home viitattaisiin edellisen esimerkin tiedostoon seuraavalla tavalla **./ijh/file1**. Piste hakemistorakenteen alussa viittaa siis käyttäjän nykyiseen sijaintiin, kaksi pistettä taas viittaisi yhden hakemistotason korkeammalle käyttäjän nykyisestä sijainnista. (Oksanen 1992, 38) Samalla tapaa jokaisessa hakemistossa on aina kaksi tiedostoa, jotka viittaavat nykyiseen hakemistoon ja yhtä tasoa korkeampaan. Yhdellä pisteellä merkitty tiedosto on hakemisto jossa käyttäjä sillä hetkellä on, ja kahdella pisteellä merkitty tiedosto tarkoittaa yhtä tasoa ylempänä hakemistorakenteessa. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 59)

Unixin tiedostojärjestelmässä käytettävät tiedostoresurssityypit voidaan jakaa kolmeen alaryhmään, levypohjaisiin, verkkopohjaisiin ja virtuaalisiin. Levypohjaisilla tarkoitetaan järjestelmään fyysisesti kytkettävää levylaitteistoa, nauhalaitteistoa, tai vaikkapa USB-muistia. Verkkopohjaisilla taas tarkoitetaan tietoliikenneohjelmistojen kautta etäyhteyden ottamista muualla jaettuihin levylaitteistoihin. Virtuaalinen tiedostoresurssi tarkoittaa tiedon tallentamista ilman varsinaisia fyysisiä tallentimia, esimerkiksi käyttämällä keskusmuistista lohkaistua väliaikaista tallennustilaa. Käyttäkseen tiedostoresurssia järjestelmässä, täytyy se ensin kiinnittää (mount) hakemistorakenteessa johonkin hakemistoon, tai juurihakemistoon. (Love & Merlino & Zimmerman & Reed & Weinstein 2005, 53 ja 77)

Unixin tiedostotyytit voidaan jakaa kolmeen perustyyppiin, hakemistoihin, tavallisiin tiedostoihin ja erikoistiedostoihin. Peruskäyttäjälle riittää yleensä tavallisten tiedostojen ja hakemistojen osaaminen, mutta järjestelmässä syvemmällä työskentelevän tulee osata myös erikoistiedostot, joihin kuuluu muun muassa erilaiset laitetiedostot, linkkitiedostot sekä prosessien väliseen liikennöintiin tarkoitettut tiedostot. (Oksanen 1992, 36 ja Love & Merlino... 2005, 63)

Kuviossa 9. on Kelan Unix-ympäristössä suoritettu ”ls -l” komento, jolla listataan työhakemiston sisältö laajemmalla tietotasolla. Kuvaan on rajattu tiedostolistauksesta kuitenkin ainoastaan osa hakemiston tiedostoista.

```

1.
  2.  3.  4.  5.  6.  7.  9.
-rwxr-x--- 1 NA03J  NA9      547 Jan 12  2005 u.map.backup
-rw-r--r-- 1 NA03J  SYS     9152 Sep 20  09:56 utmpx
drwxrwxr-x 2 NA03J  OMV      288 Oct 13  2010 wbem
-rwxrwx--- 1 NA03J  SYS    15520 Jan 25  1998 yylex.c
-rwxrwx--- 1 NA03J  SYS   22508 Jan 25  1998 yyparse.c
drwxrwxr-x 3 NA03J  OMV      288 Oct 13  2010 zWebSphereOEM
drwxr-xr-x 2 NA03J  NA9      256 Mar 14  1996 zoneinfo
drwxrwx--- 2 NA03J  OMV      288 Oct 13  2010 zosmf
AS:/AS/etc>

```

Kuvio 9. Kelan /AS/etc hakemistossa annettu ls -l komento.

Tarkastellaan kuvion 9. numeroinnin avulla tavallisimpia tietoja, joita Unixin tiedostolistauksesta saadaan selville.

1. Ensimmäinen merkki määrittelee tiedostotyyppin. Merkki ”d” tarkoittaa hakemistoa, ”-” merkki taas tavallista tiedostoa. Muut merkit kuvaavat erikoistiedostoja. Erikoistiedostoista yleisin on linkkitiedosto ”l”, jolla on mahdollista tehdä oikotie mihin tahansa tiedostoon järjestelmässä.
2. Seuraavat 9 merkkiä määrittelevät tiedoston käyttöoikeudet ja suojauksen. Nämä 9 merkkiä on jaettu kolmeen osaan siten, että ensimmäiset kolme kuvaavat tiedoston omistajan oikeuksia, seuraavat kolme määrittelevät tiedoston käyttäjäryhmään kuuluvien käyttäjien oikeudet ja viimeiset kolme kaikkien muiden käyttäjien oikeudet. Kolmen merkin ryhmä on jaettu siten, että ensimmäisen merkin ollessa ”r” kuvaa se luku-oikeutta tiedostoon, toisen merkin ”w” kirjoi-

tusoikeutta ja kolmannen ”x” suoritusoikeutta. Mikäli merkinä missä tahansa kohdassa on ”-”, tarkoittaa se, ettei kyseistä oikeutta ole. Esimerkiksi ”rwxr-xr--” määritteli tiedoston omistajalle kaikki oikeudet, tiedoston käyttäjärhymälle luku- ja suoritusoikeudet ja kaikille muille lukuoikeuden tiedostoon.

3. Tällä kuvataan kuinka monta yhteyttä järjestelmässä on tiedostoon. Jokainen linkkitiedosto, joka osoittaa alkuperäiseen tiedostoon, nostaa yhteyksien määrää.
4. Tämä on tiedoston omistajan käyttäjätunnus.
5. Tässä kuvataan tiedoston omistava käyttäjärhymä. Oletuksena käyttäjärhymä on tiedoston omistajan käyttäjätunnukselle määritelty ensisijainen ryhmä.
6. Seuraava luku kertoo tiedoston todellisen koon tavuina.
7. Tässä on kuvattu tiedoston viimeisin muokkausajankohta, kuukausi, päivä ja vuosi. Tai mikäli tiedostoa on muokattu kuluvana vuonna, vuosi on korvattu kellonajalla.
8. Viimeisenä tietona on tiedoston tai hakemiston nimi.

## 4.2 Käyttöjärjestelmä z/OS

z/OS on IBM:n System z -sarjan suurtietokoneille tarkoitettu käyttöjärjestelmä. Sen kehityshistoria ulottuu 1960-luvulle saakka, z/OS käyttöjärjestelmän edeltäviä versioita ovat OS/390-, OS/MVS- ja OS/360-käyttöjärjestelmät (Haikala & Järvinen, 2004, 44). Jokainen näistä käyttöjärjestelmistä on perustunut siihen ajatukseen että ne soveltuvat kaikkiin saman sarjan suurtietokoneisiin. Tällöin asiakkaan on ollut mahdollista rakentaa tietojärjestelmänsä luottaen siihen, että ne toimivat edelleen vaihdettaessa suurtietokoneita saman sarjan uudempaan malliin. Samankaltaista jatkuvuuden mahdollistavaa järjestelmää ei aikaisemmin ollut toteutettu. 360-suurtietokoneiden ja OS/360-käyttöjärjestelmän myötä alkoi käytännössä IBM:n vuosikymmeniä kestänyt voittokulku tietotekniikassa. (Rojas & Hashagen, 2000, 203–204) Edellä kuvatun kaltaisesti myös Kela on rakentanut vuosikymmenten ajan tietojärjestelmiään IBM:n suurtietokoneiden ja käyttöjärjestelmien päälle. Kela siirtyi jo vuonna 1960, toisena kansallisena eläkelaitoksena maailmassa, käyttämään tietokoneita eläkkeiden maksamiseen (Häggman, 1997, 121).

Johtuen käyttöjärjestelmän pitkästä historiasta ja siitä että se on vuosikymmenten saatossa aina ollut hieman eriytynyt muista tietokonejärjestelmistä, sen terminologia on

muodostunut yleisistä standardeista jonkin verran poikkeavaksi. Tämä aiheuttaa välillä hankaluuksia, tuttujen termien ja lyhenteiden loistaessa poissaolollaan, tai jopa tarkoittaen aivan jotain muuta kuin mihin on tottunut.

z/OS käyttöjärjestelmä suoritetaan suurtietokoneen keskusprosessorilla ja se sijaitsee aina käynnistyttyään keskusmuistissa (IBM z/OS Basic Skills Information Center I, 2010, 2). Kuviossa 10. esitellään z/OS käyttöjärjestelmän käyttämät laitteistoresurssit.



Kuvio 10. z/OS käyttöjärjestelmän käyttämät laitteistoresurssit (IBM z/OS Basic Skills Information Center, 2010, 2).

z/OS on moniajokäyttöjärjestelmä, joka pystyy käyttämään yhtä aikaa hyväkseen kahta tai useampaa prosessoria, jotka jakavat yhteiset laiteresurssit (IBM z/OS Basic Skills Information Center I, 2010, 3). Moniajokäyttöjärjestelmäksi kutsutaan käyttöjärjestelmää, jonka keskusmuistissa on mahdollista olla monta prosessia yhtä aikaa, jolloin suoritettavaksi voidaan valita mikä tahansa suoritukseen valmis prosessi (Haikala & Järvinen, 2004, 31).

Toimintavarmuus on yleisesti yksi z/OS-käyttöjärjestelmien vahvuuksista. Toimintavarmuutta on mahdollisuus lisätä entisestään, liittämällä yhteen useita z/OS-käyttöjärjestelmää käyttäviä järjestelmiä. Tällaista z/OS-järjestelmien rypästä, jossa tietoa voidaan käsitellä järjestelmien kesken kulloisenkin järjestelmän kuorman

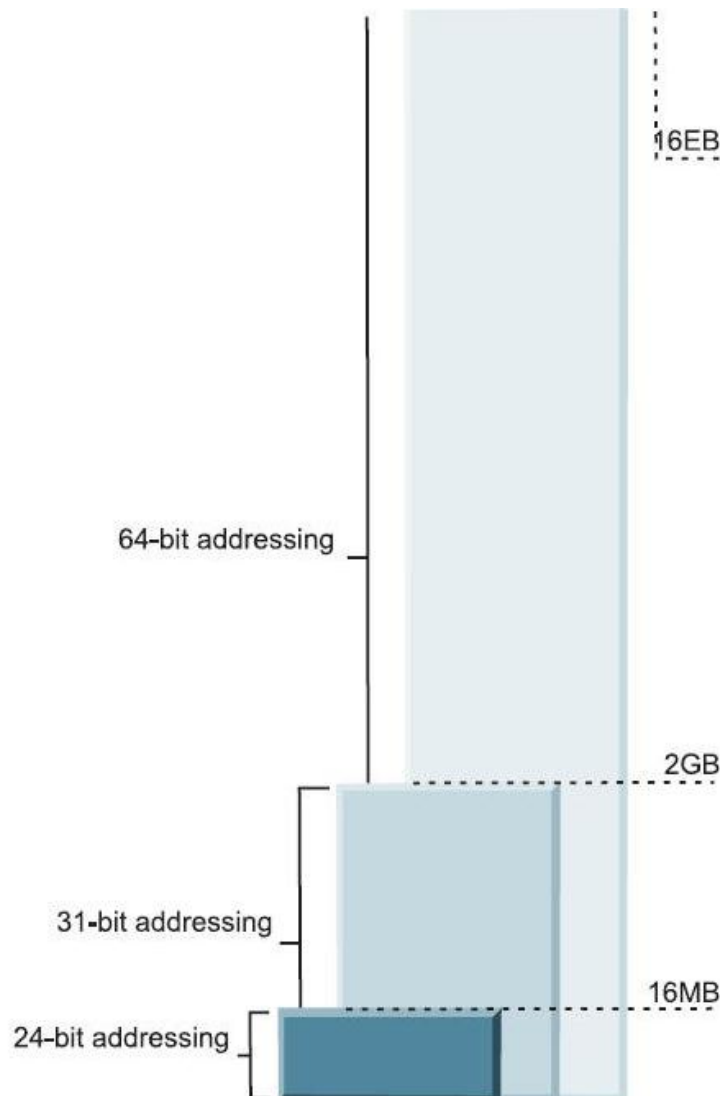
sallimissa rajoissa, siten ettei suoritustehokkuus tai tiedon eheys kärsi, kutsutaan termillä Parallel Sysplex (IBM z/OS Basic Skills Information Center III, 2008, 42). Toimintavarmuuden lisäksi Parallel Sysplex mahdollistaa järjestelmäkokonaisuuden dynaamisen hallinnan, eli järjestelmiä voidaan lisätä tai poistaa ilman palvelukatkoja (IBM z/OS Basic Skills Information Center III, 2008, 44). Oikein toteutettuna Parallel Sysplex'in toiminta olisi hyvin lähellä neljännen sukupolven hajautetun käyttöjärjestelmän erityispiirteitä ja täyttäisi tuntumattomuuden määritelmän.

IBM z/OS Basic Skills Information Center I (2010, 101-106) mukaan z/OS-järjestelmien toimiminen Parallel Sysplex:inä tuo mukanaan seuraavia etuja:

- Kaikki järjestelmässä toimivat kriittiset palvelut on mahdollista kahdentaa.
- Järjestelmän skaalautuvuus. Suoritustehokkuutta on mahdollista nostaa lisäämällä järjestelmiä Parallel Sysplex:iin.
- Dynaaminen kuorman jakaminen Parallel Sysplex:iin kuuluvien järjestelmien välillä.
- Helppokäyttöisyys. Monet z/OS:n mukana tulevat apuvälineet on tarkoitettu tehostamaan Parallel Sysplex:ssä toimimista.
- Yksi hallittava järjestelmä, vaikka Parallel Sysplex koostuisi useasta suurkoneesta ja jopa 32 z/OS-järjestelmästä.
- Järjestelmätuki aikaisemmille versioille. Ainakin kahta tuoreinta järjestelmätasoa tuetaan ja niitä voidaan ajaa rinnakkain Parallel Sysplex:ssä.
- Ohjelmistojen tuki. z/OS-järjestelmässä toimivien ohjelmistojen pitäisi toimia myös Parallel Sysplex:ssä.
- Ongelmista toipuminen. Kriittinen tieto peilataan järjestelmien välillä, mahdollistaen palveluiden nopean toipumisen yksittäisten järjestelmien tai vaikka yhden palvelimen vikaantuessa.

Virtuaalinen muisti on yksi z/OS käyttöjärjestelmän keskeisimmistä toimintaperiaatteista. Virtuaalisella muistilla järjestelmässä saadaan käytettäväksi enemmän muistiavaruutta, kuin siinä todellisuudessa onkaan. Vain ohjelmien aktiiviset osat, joita todella tarvitaan, ladataan keskusmuistiin. Passiivisena olevat ohjelmien osat säilötään apumuistiin (auxiliary memory). Virtuaalisen muistin määrä on sidoksissa todelliseen keskusmuistiin ja riippuu käytettävistä ohjelmistoista. z/OS rakentuu virtuaalisen muistin osoiteavaruuksien ympärille. Jokainen käyttäjä saa oman määritellyn osoite- sekä muis-

tiavaruuden käyttöönsä. Näitä on mahdollista jakaa järjestelmässä toisten käyttäjien ja ohjelmien kanssa, tai sitten muistiavaruus voidaan määritellä käyttäjän omaksi eristetyksi tietoturvalliseksi muistialueeksi. z/OS:n osoiteavaruus on käsitteenä samantapainen kuin prosessit Unix:ssa, molemmat varaavat tunnistettavissa olevan muistialueen tietyn ohjelman, tai ohjelman osan käyttöönsä. Unix prosessit yksilöidään (PID) prosessitunnuksilla ja vastaavasti z/OS:n osoiteavaruuden prosessit (ASID) tunnuksilla. Ohjelmien suorittaminen z/OS käyttäjärjestelmässä voidaan toteuttaa 24-, 31-, tai 64-bittisessä osoiteavaruudessa, tai vaihdella tarvittaessa suorittamisen aikana näiden välillä. Kuviossa 11. on esimerkki osoiteavaruudesta sekä muistinkäytön mahdollisuuksista. (IBM z/OS Basic Skills Information Center, 2010, 1 ja 6)



Kuvio 11. z/OS käyttäjärjestelmän 64-bittinen osoiteavaruus mahdollistaa jopa 16 ek-satavun kokoisen muistiavaruuden käyttämisen (IBM z/OS Basic Skills Information Center I, 2010, 15).



z/OS-käyttöjärjestelmän kiinteänä osana toimii UNIX System Services (USS), joka on z/OS:ään integroitu Unix-käyttöjärjestelmä. USS noudattaa rekisteröityjen Unix-standardien (XGP4 ja UNIX 95) periaatteita. (z/OS V1R12.0 UNIX System Services Planning, 2010, 1). USS mahdollistaa Unix-pohjaisten sovellusten ajamisen z/OS-käyttöjärjestelmällä varustetulla suurtietokoneella. USS ja z/OS:n perinteinen MVS-puoli pystyvät toimimaan yhdessä siten, että molempien järjestelmien hyvät puolet saadaan valjastettua käyttöön. Aivan ongelmattomasti niiden yhteistoiminta ei kuitenkaan ole, johtuen mm. erilaisista merkkikoodistoista. Eroavaisuuksiin ja yhteensovittamisen ongelmiin palataan tarkemmin edempänä tekstissä.

Tietokonejärjestelmät, jonka kaltaisia ei aikaisemmin ole käyttänyt, tuntuvat käyttäjistään helposti todella vaikeilta hallita. Tavallisilla käyttäjillä on kuitenkin yleensä z/OS:n alla vain yksi tai kaksi ohjelmistoa, jotka he käynnistävät käyttöliittymän valikoiden kautta ja joita he käyttävät työtehtäviensä suorittamiseen. Käyttämisen jälkeen he kirjautuvat ulos järjestelmästä, eikä syvällistä osaamista vaadita. Perehtymistä ja järjestelmän syvällistä ymmärtämistä tarvitaan ainoastaan pääkäyttäjiltä, joita suurtietokoneympäristössä kutsutaan *system programmer* -nimikkeellä. Heidän tehtäviinsä kuuluu käyttöjärjestelmän sekä ohjelmistojen hallinta, asentaminen ja muokkaaminen. Peruskäyttäjien ja pääkäyttäjien välinen ero z/OS-järjestelmässä on samanlainen kuin missä tahansa muussa käyttöjärjestelmässä. Ensin mainitut osaavat käyttää järjestelmää sen verran, että pystyvät suorittamaan työtehtäviensä, jälkimmäisten työtehtävän ollessa järjestelmän hallinta itsessään.

#### **4.2.1 z/OS tiedostojärjestelmät**

z/OS käyttöjärjestelmän MVS-tiedostojärjestelmä perustuu dataset -tiedostojen käyttämiseen. Dataset-tiedostolla tarkoitetaan tiedostoa, joka koostuu yhdestä tai useammasta tietueesta. z/OS käyttöjärjestelmän dataset-tiedostojen käsittely pohjautuu tietueiden käsittelyyn, toisin kuin esimerkiksi Unix- ja Windows-käyttöjärjestelmissä jossa tiedostot käsitellään tavuina. Dataset tiedostolle täytyy aina ennen sen käyttämistä allokoida tarvittava tilamäärä, joko manuaalisesti tai sitten erillisen automatiikan avulla. Tyypillisiä attribuutteja joita dataset tiedostolle täytyy määritellä ovat tietuetyyppi (record format), loogisen tietueen maksimi pituus (maximum logical record length) ja

maksimi lohkokoko (maximum block size). Dataset voi siis olla vaikkapa yksittäinen ohjelmatiedosto, tai sitten kirjastollinen ohjelmatiedostoja. Yleisimmät dataset tiedostotyypit ovat peräkkäistiedosto, kirjastotiedosto joka sisältää kirjaston jäsentiedostoja (member) ja VSAM-tiedosto (Virtual Storage Access Method) jota ei voi editoida normaaleilla menetelmillä, vaan se on tarkoitettu ohjelmistojen käytettäväksi. Dataset-tiedostojen nimet voivat muodostua maksimissaan 44 suurakkesta. Nimet on jaettu tasoihin siten, että yksi taso voi olla maksimissaan 8 merkkiä pitkä. Tasot erotetaan toisistaan pisteellä, järjestelmän ylin taso on yleensä suojattu. Yksittäisen dataset:in löytääkseen tulee tietää sen nimen lisäksi myös levylaitetyyppi sekä levyn nimi, tai mikäli tiedosto on lisätty MVS-tiedostojärjestelmän katalogiin, riittää etsimiseen pelkkä nimi. (IBM z/OS Basic Skills Information Center I, 2010, 35 ja 36.) Dataset -tiedostot tallennetaan joko DASD (Direct Access Storage Device) levylaitteistolle, tai kasettijärjestelmään (IBM z/OS Basic Skills Information Center I, 2010, 37).

z/OS käyttöjärjestelmä käyttää dataset -tiedostoissaan EBCDIC(Extended Binary Coded Decimal Interchange) -merkkikoodistoa (IBM, z/OS Basic Skills Information Center II, 2010, 9). IBM kehitti EBCDIC -merkkikoodiston reikäkorttijärjestelmien pohjalta OS/360-järjestelmiään varten. Lähes yhtä aikaa sen kanssa kehitettiin ASCII (American Standard Code for Information Interchange) -merkkikoodisto, josta tuli alan standardi ja jota käyttämään siirtyivät kaikki toimijat IBM:ää lukuun ottamatta. Koska merkkikoodistot eivät olleet toistensa kanssa täysin yhteensopivia, aiheutui tästä tiedonsiirroissa eri järjestelmien välillä käännoongelmia. (Ceruzzi, 2003, 152.) Nykyään käyttäjien ei yleensä enää tarvitse huolehtia merkkikoodikäännoksistä, vaan järjestelmät suorittavat käännookset automaattisesti (IBM, z/OS Basic Skills Information Center II, 2010, 9). EBCDIC- ja ASCII-merkkikoodistojen välisiä eroja havainnollistetaan kuvioissa 12. ja 13.

<b>Merkki</b>	<b>EBCDIC</b>	<b>ASCII</b>
A	11000001 (x'C1')	01000001 (x'41')
B	11000010 (x'C2')	01000010 (x'42')
a	10000001 (x'81')	01100001 (x'61')
1	11110001 (x'F1')	00110001 (x'31')
välilyönti	01000000 (x'40')	00100000 (x'20')

Kuvio 12. Merkkikoodistot muodostavat merkit yleensä eri binäärikoodilla (IBM, z/OS Basic Skills Information Center II, 2010, 9).

	EBCDIC	ASCII
Alhaisin arvo koodistossa:	välilyönti	välilyönti
	välimerkki	välimerkki
	pienaakkoset	numerot
	suuraakkoset	suuraakkoset
Korkein arvo koodistossa:	numerot	pienaakkoset

Kuvio 13. Binäärikoodien arvoasteikoissa pienaakkoset ja numerot ovat EBCDIC- ja ASCII-merkkikoodistoissa toisin päin (IBM, z/OS Basic Skills Information Center II, 2010, 10).

Dataset tiedostot ovat siis varsin erilaisia, kuin mihin Windows- ja Unix-käyttöjärjestelmiä käyttäneet ovat tottuneet. Hakemistorakenteiden puuttuminen on yksi suurimmista eroista. Kirjasto dataset -tiedostot ovat lähinnä hakemistoja, mutta niissä voi sijaita ainoastaan jäsentiedostoja, eli puumaista hakemistorakennetta ei perinteisessä MVS-tiedostojärjestelmässä muodostu. Toinen mahdollisia ongelmia aiheuttava asia on dataset -tiedostojen tilan määrän varauksen tekeminen jo luomisvaiheessa. Arviointi on vaikeaa jolloin tiedostoista tulee helposti joko suhteettoman suuria tai sitten liian pieniä. Tästä syystä z/OS käyttöjärjestelmän kanssa käytetään hyvin usein apuna Data Facility Storage Management Subsystem (DFSMS) tilanhallintajärjestelmää. DFSMS avulla monet tilanhallintaan liittyvät toimet voidaan automatisoida (IBM z/OS Basic Skills Information Center I, 2010, 53). Tiedoston tilan loppumisesta kertova ilmoitus on silti yksi yleisimmistä virheilmoituksista käyttöjärjestelmässä.

Muista käyttöjärjestelmistä z/OS:n pariin siirtyneelle tilan arviointia vaikeuttaa entisestään erilaiset tilanvarausyksiköt. Koska tiedostot eivät perustu tavujen käyttämiseen, tiedostojen kokoja ei mitata tavuissa, kuten yleisesti on totuttu. Yleisimmät tilanvarausyksiköt ovat *cylinder* ja *track* (IBM, DFSMSdfp Storage Administration, 2010, 20). Pikakatsaus dataset-tiedostojen tilanvarausyksiköihin suoritetaan kuviossa 14.



Kuvio 14. z/OS levylaitteiden tilanvarausyksiköt (IBM z/OS Basic Skills Information Center, 2010, 37).

MVS-tiedostojärjestelmä ei ole ainoa tiedostojärjestelmä, mitä z/OS-käyttöjärjestelmässä käytetään, vaan USS-ympäristössä tiedostoresurssien hallintaan käytetään Unix-tiedostojärjestelmää. USS-ympäristön Unix-tiedostojärjestelmässä täytyy käyttää tiedostoresurssit ensin määrittellä MVS-tiedostojärjestelmän puolella ja sen jälkeen kiinnittää (mount) tiedostoresurssi tiedostojärjestelmän juureen, tai tyhjään hakemistoon. (IBM z/OS Basic Skills Information Center I, 2010, 55). Tämän jälkeen juureen tai hakemistoon voidaan luoda taikka kopioida tiedostoja sekä hakemistoja, siihen kiinnitetyn tiedostoresurssin tilanvarauksen sallimissa rajoissa. Kiinnitettyjä tiedostoresursseja voi olla hakemistorakenteessa useassa tasossa. Tällöin alemmalle tasolle hakemistorakenteessa kiinnitetyn hakemiston sisällä olevat tiedostot eivät käytä ylemmän tason kiinnityksen tilanvarausta.

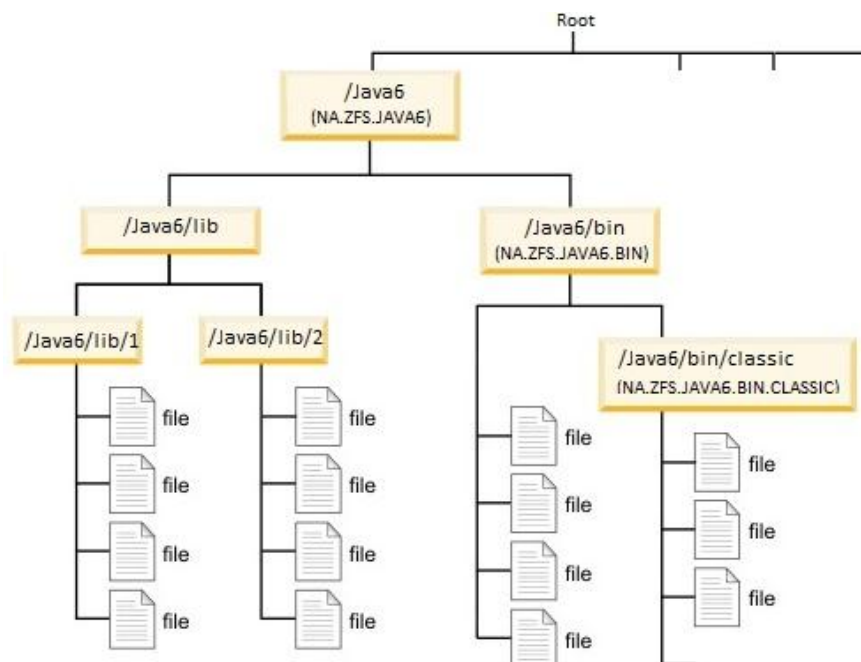
Tiedostojen nimeämiseen, etsimiseen, merkistökoodiin ja hakemistorakenteisiin liittyvissä säännöissä z/OS:n Unix-tiedostojärjestelmää koskevat yleiset

Unix-tiedostojärjestelmän lainalaisuudet. MVS-tiedostojärjestelmässä tiedostojen etsintää helpottava katalointi ei toimi USS-ympäristössä.

z/OS concepts (2010, 55) manuaalin mukaan seuraavia tiedostoresurssityyppejä on mahdollista käyttää USS-ympäristön Unix-tiedostojärjestelmässä:

- zSeries File System (zFS), on tällä hetkellä suositeltavin kiinnitettävä fyysinen tiedostoresurssi, koska se tukee tehokkaampaa VSAM-tiedonsaanti menetelmää.
- Hierarchical file system (HFS), on kiinnitettävä fyysinen tiedostoresurssi, jonka tehokkaampi ja monipuolisempi zFS on alkanut syrjäyttää.
- z/OS Network File System (z/OS NFS), mahdollistaa ulkopuolisen Unix-tiedostojärjestelmän käyttämisen TCP/IP-yhteydellä.
- Temporary file system (TFS), tilapäinen tiedostoresurssi, jossa tieto säilytetään muistijärjestelmässä eikä varsinaisella levylaitteella.

Kuviossa 15. on zFS-tyyppiset dataset-tiedostot NA.ZFS.JAVA6, NA.ZFS.JAVA6.BIN ja NA.ZFS.JAVA6.BIN.CLASSIC kiinnitetty hakemistorakenteeseen er. Juuren alla sijaitsevaan hakemistoon /Java6 on kiinnitetty tiedostoresurssiksi NA.ZFS.JAVA6, sen alapuolella olevaan hakemistoon /Java6/bin on kiinnitetty NA.ZFS.JAVA6.BIN ja sen alapuolella olevaan /Java6/bin/classic –hakemistoon NA.ZFS.JAVA6.BIN.CLASSIC.

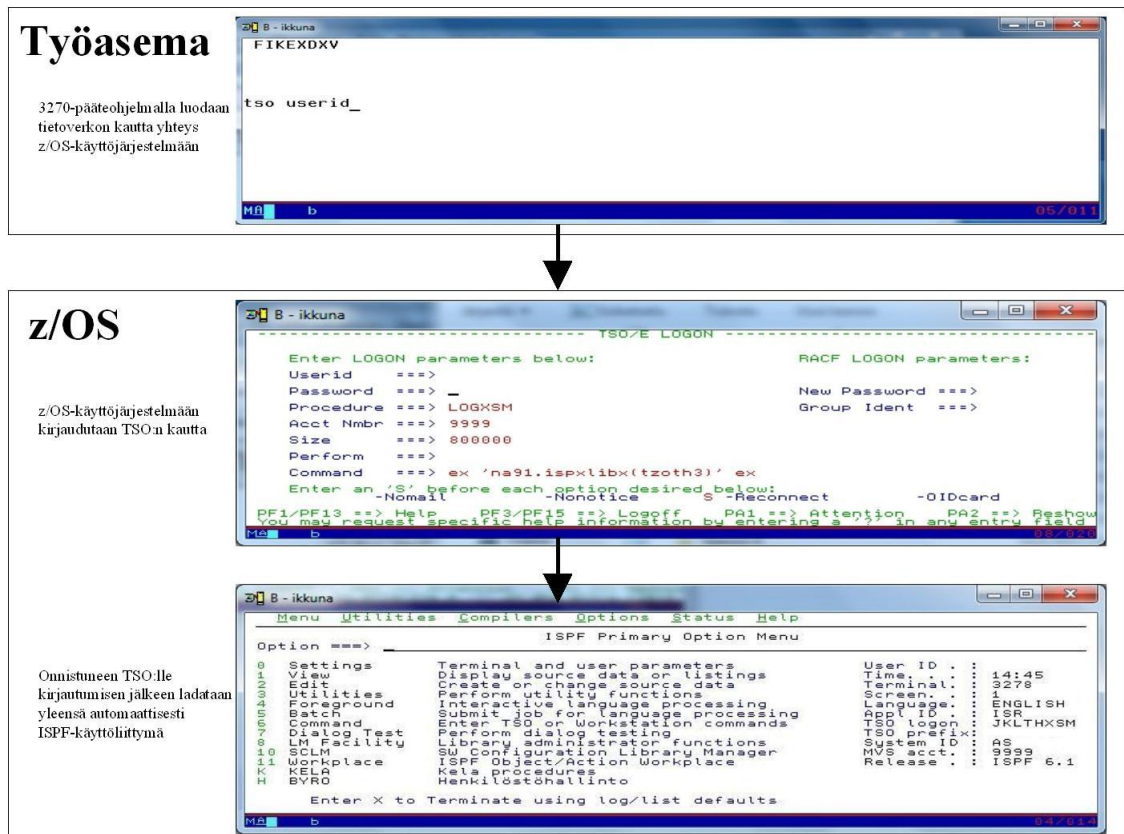


Kuvio 15. USS-ympäristöön kiinnitettyjä zFS-tiedostoresursseja.

Terminologiassa dataset tarkoittaa eri asiaa, riippuen siitä työskennelläänkö MVS-tiedostojärjestelmän vai Unix-tiedostojärjestelmän kanssa. MVS:n alla dataset on synonyymi tiedostolle, mutta Unix:ssa dataset ei ole tiedosto, vaan zFS:n tai HFS:n kaltaisen tiedostoresurssi.

#### 4.2.2 z/OS käyttöliittymät

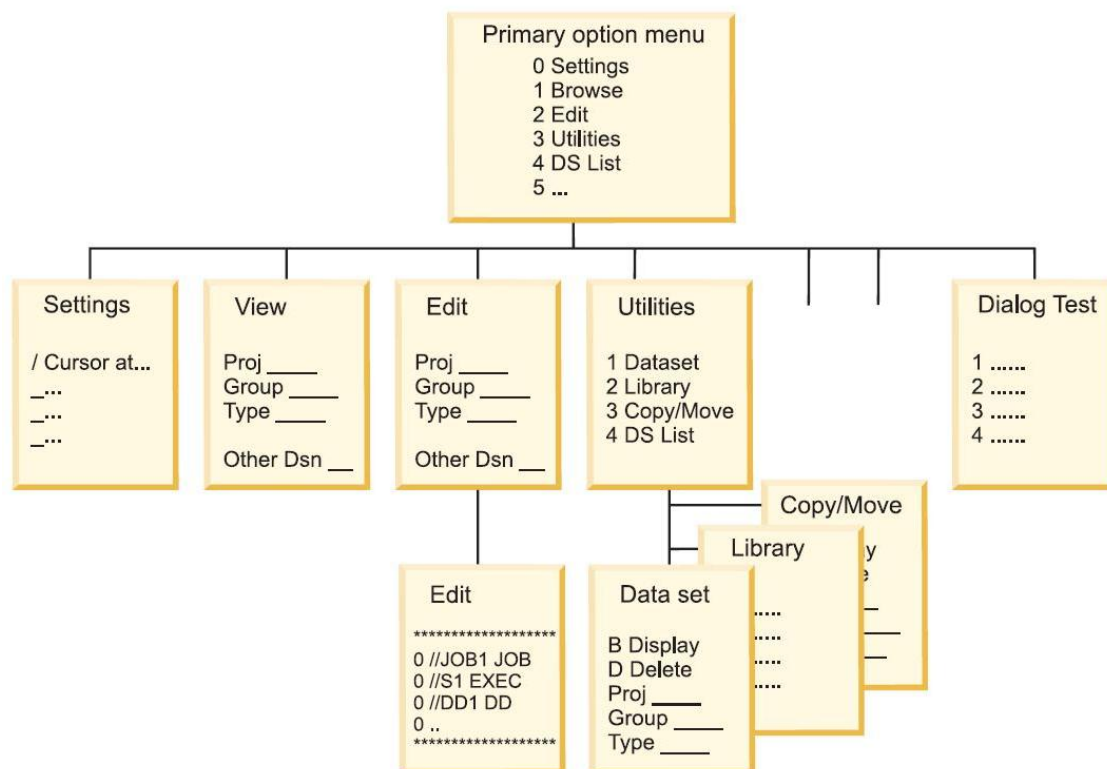
Tavalliselle käyttäjälle z/OS käyttöjärjestelmän historia näkyy parhaiten käyttöliittymän kautta. Nykyään lähes kaikkia järjestelmiä on totuttu käyttämään graafisten, helppokäyttöisten ja suorastaan käyttäjänsä kädestä pitäen ohjaavien käyttöliittymien kautta. z/OS käyttöjärjestelmän käyttöliittymässä näkyy vielä aika jolloin ei ollut muita tietokoneita kuin suurtietokoneet, joihin otettiin yhteyttä ja joita käytettiin päätelaitteiden kautta. Nykyään ei ole enää käytössä päätelaitteita, mutta työasemalle asennettava 3270-pääteohjelma emuloi sellaista ja mahdollistaa yhteysistunnon avaamisen z/OS käyttöjärjestelmään. Yhteysistunto muodostetaan käyttäen käyttäjätunnusta ja salasanaa Time Sharing Option (TSO) -käyttöliittymäpohjaan, jonka päällä toimii yleensä valikopohjainen Interactive System Productivity Facility (ISPF) -käyttöliittymä (IBM z/OS Basic Skills Information Center I, 2010, 59). TSO/ISPF -käyttöliittymä on teksti-, tai merkkipohjainen ja sitä käytetään pääsääntöisesti pelkästään näppäimistön avulla. Kuviossa 16. on esimerkki Kelassa tapahtuvasta yhteysistunnon muodostamisesta ja TSO/ISPF -käyttöliittymästä.



Kuvio 16. Yhteyden muodostus ja käyttöliittymän avaaminen z/OS-käyttöjärjestelmään Kelassa.

Vaikka TSO:lle muodostetun yhteyden jälkeen yleensä heti ladataan ISPF-käyttöliittymä, on TSO itsessään yksinkertainen komentotulkki, jonka kautta on mahdollista käyttää z/OS käyttöjärjestelmää. TSO-komentotulkki muistuttaa hieman vanhan DOS-käyttöjärjestelmän komentokehoitetta. Se on varsin yksinkertainen ja rajoittunut, josta syystä z/OS:ää käytetään lähes aina ISPF:n avulla. (IBM z/OS Basic Skills Information Center I, 2010, 60-61.)

ISPF:n valikot sisältävät yleisimmät toiminnot, joita z/OS käyttäjät tarvitsevat. Valikoiden kautta avataan tekstien luku- ja muokkaus-ohjelmat, tiedostojen hallintatyökalut sekä kaikki z/OS:n alle asennetut aliohjelmistot. Valikoiden käyttäminen tapahtuu näppäimistöllä. (IBM z/OS Basic Skills Information Center I, 2010, 62) ISPF:n valikoita on mahdollista muokata haluamaansa muotoon, jolloin käyttäjällä on mahdollisuus käynnistää usein käyttämänsä ohjelmistot helposti. Lisäksi funktionäppäinten alle on mahdollista lisätä pikakomentoja, jotka helpottavat päivittäistä käyttöä. ISPF:ssä oletuksena oleva valikkorakenne esitellään kuviossa 17.

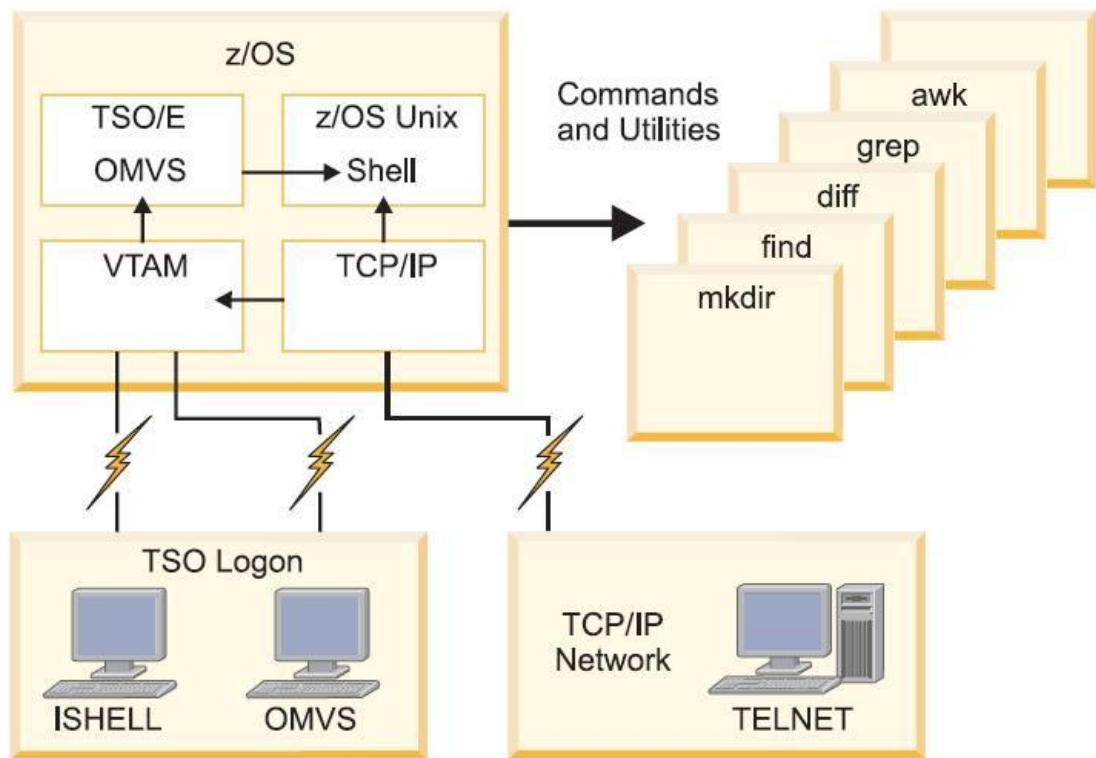


Kuvio 17. ISPF:n oletusnäytön valikkorakenne (IBM z/OS Basic Skills Information Center I, 2010, 64).

USS-ympäristön käyttämiseen on olemassa muutamia vaihtoehtoja. TSO/ISPF-yhteyden muodostamisen jälkeen käyttäjä voi käynnistää joko ISPF-valikoihin perustuvan ISPF Shell'in (ISHELL), tai perinteisemmän näköisen Unix komentotulkin(Shell) OMVS -komennolla. Kolmas vaihtoehto on suora yhteyden muodostus z/OS Unixin komentotulkkiin, käyttämällä Telnet- tai Rlogin-ohjelmaa. (IBM z/OS Basic Skills Information Center I, 2010, 73)

z/OS Unixin komentotulkki perustuu POSIX standardiin ja se pohjautuu UNIX System V -komentotulkista, joskin siinä on ominaisuuksia myös UNIX Korn -komentotulkista. Komentotulkki on kaikkein monipuolisin käyttöliittymä USS-ympäristön hallintaan. Sen kautta voi luoda ja kutsua komentojonotiedostoja (shell script), sekä käyttää yleisimpiä Unixin apuvälineitä ja komentoja. Mikäli Unix-käyttöjärjestelmät ovat käyttäjälle tuttuja, on z/OS Unix komentotulkki paras tapa operoida USS-ympäristössä. (IBM z/OS Basic Skills Information Center I, 2010, 73-74.) z/OS Unixin komentotulkin toiminta esitellään kuviossa 18.





Kuvio 18. z/OS Unix komentotulkki (IBM z/OS Basic Skills Information Center I, 2010, 73).

Toinen mahdollisuus hallita USS-ympäristöä on käyttää ISPF-käyttöliittymän ISHELL-työkalua. Se sopii parhaiten henkilöille, joilta ei löydy kokemusta Unixin käyttämisestä, mutta joille TSO/ISPF on tutumpi (IBM z/OS Basic Skills Information Center I, 2010, 74). ISHELL on kuitenkin varsin rajoittunut työkalu ja se soveltuu hyvin ainoastaan USS-ympäristön tiedostojen muokkaamiseen ja hallintaan. Muu operointi USS:n alla on käytännössä suoritettava komentotulkkia käyttäen. Kuviossa 19. on ruutukopio ISHELL-työkalun käyttöliittymästä.

```

Command ==> Directory List
Select one or more files with / or action codes.
EUID=13096165 /AR/var/
Type Perm Changed-EET-2EET Owner -----Size Filename
- Dir 771 2012-08-29 10:13 NB 1568 Printsrv
- Dir 755 2012-04-29 11:18 NB 1504 .
- Char 777 2012-04-29 11:18 NB 0 CEAServer
- Dir 755 2012-03-11 11:49 NB 256 javaold
- Dir 775 2011-11-22 08:38 NB 704 imwebsrv
- Dir 777 2011-07-12 14:01 NB 8192 javabatch
- Dir 777 2010-12-12 13:42 NA 320 in
- Dir 755 2010-12-12 10:49 NA 576 ..
- Dir 750 2010-12-10 15:31 NA 416 log
- Dir 777 2010-12-10 14:50 NA 288 internet
- File 666 2010-12-10 14:20 NA 2786 kelasto\testi\TVS\TESTI.PDF
- File 666 2010-12-10 14:20 NA 2786 kelasto\testi\TESTI.PDF
- File 755 2010-12-10 14:20 NA 995 .testi
- Dir 777 2010-12-10 14:06 NA 256 wasdata
- Dir 777 2010-12-10 14:06 NA 256 sambaroot
- Dir 755 2010-10-13 20:08 NB 320 wbem
- Dir 775 2010-10-13 19:44 NB 512 ocsf
- Dir 775 2010-10-13 17:23 NB 288 zWebSphereOEM
- Dir 755 2010-10-13 16:56 NB 288 zosmf
- Dir 755 2010-09-06 08:56 NB 288 skrb
- Dir 755 2010-09-06 08:56 NB 256 pkiserv
- Dir 755 2010-09-06 08:56 NB 256 pfa
- Dir 755 2009-06-10 09:27 NB 288 netview
- Dir 777 2009-06-07 11:32 NB 256 extranet
- Dir 755 2008-11-04 14:37 NB 288 siirrot

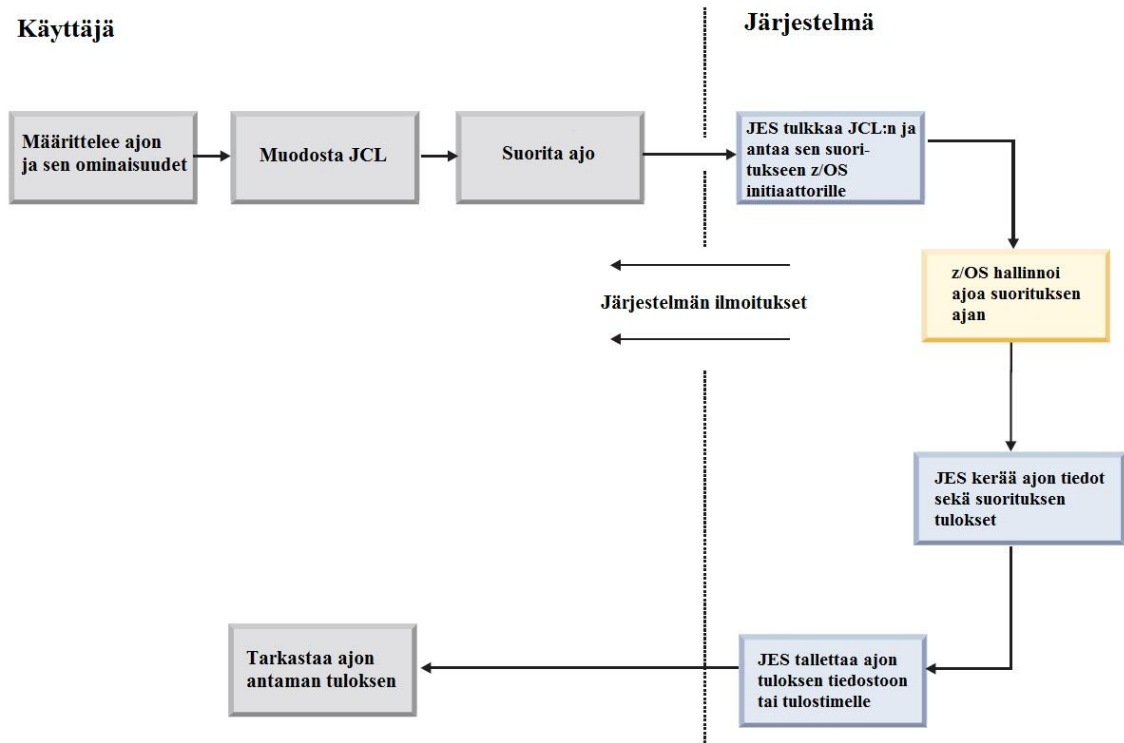
```

Kuvio 19. ISHELL näkymä Kelassa hakemistosta /AR/var/.

### 4.2.3 Eräajojen ja töiden hallinta z/OS-käyttöjärjestelmässä

Eräs tärkeimmistä toimintaperiaatteista z/OS-käyttöjärjestelmässä on eräajojen hallinta (IBM z/OS Basic Skills Information Center I, 2010, 79). Eräajoilla tarkoitetaan ajastettuja, käytettävissä olevien resurssien mukaan suoritettavia ohjelmia tai ohjelmaryhmiä, joiden ajaminen tapahtuu automaattisesti sen jälkeen kun niiden suoritusparametrit on määriteltä. Eräajoina suoritetaan yleensä organisaation kaikkein tärkeimmät ydintoinnot, esimerkiksi Kelassa etuuspäätökset ja niistä asiakkaille lähtevät kirjeet. Eräajojen toiminnan määrittelee komentosarjakieli Job Control Language (JCL). JCL:ssä ajolle määritellään omistaja, käytettävät ohjelmat, syötetiedostot, tuloksen käsittely ja säilyttäminen sekä ajankohta jolloin ajo suoritetaan (IBM z/OS Basic Skills Information Center I, 2010, 80).

z/OS-käyttöjärjestelmässä ajoja vastaanottaa, aikatauluttaa ja säilyttää Job Entry Subsystem (JES) -alijärjestelmä. Se huolehtii lisäksi ajojen jonottamisesta annettujen kriteerien perusteella, initiaattori (initiator) ohjelmien yhteistoiminnasta valittujen jonojen kanssa, ajojen tulosteiden vastaanotosta ja säilyttämisestä sekä tarvittaessa ajojen tulosteiden lähettämisestä tulostimelle tai arkistointijärjestelmään. (IBM z/OS Basic Skills Information Center I, 2010, 80) Esimerkki yksittäisen ajon suorittamisesta z/OS:n alla kuviossa 20.



Kuvio 20. Kaavio ajon suorittamisesta z/OS-käyttöjärjestelmässä (IBM z/OS Basic Skills Information Center I, 2010, 87).

IBM z/OS Basic Skills Information Center I (2010, 89) mukaan z/OS:n alla suoritettavalla ajolla on seuraavat vaiheet:

- Syöte
- Kääntäminen
- Suorittaminen
- Tulos
- Tuloksen kirjoittaminen
- Ajon poistaminen järjestelmästä

Suoritettavan ajon seurantaan, hallintaan ja tulosten tarkasteluun on mahdollista käyttää muutamia apuvälineitä. Yleisin näistä apuvälineistä on System Display and Search Facility (SDSF), joka usein hankitaan lisämaksullisena apuohjelmana z/OS käyttöjärjestelmän mukana. SDSF soveltuu monipuolisten ajojen hallintaan ominaisuuksien lisäksi käyttöjärjestelmän lokitietojen tutkimiseen, järjestelmäkomentojen antamiseen, initiaattoreiden hallintaan ja tulostimien sekä tulostustöiden hallintaan (IBM z/OS Basic Skills

Information Center I, 2010, 96). Toinen yleinen apuväline on Kelalla käytössä oleva (E)JES, joka sisältää SDSF-apuohjelmaa laajemmat ominaisuudet järjestelmän eri ominaisuuksien hallintaan. (E)JES palvelin(server) sijaitsee z/OS käyttöjärjestelmässä, mutta client sovellus toimii Windows käyttöjärjestelmässä. Tämä mahdollistaa ajojen hallinnan ja järjestelmän tilan seuraamisen myös sellaiselta henkilökunnalta, jolla ei ole taitoja työskennellä suurympäristössä. Kuviossa 21. yleisnäkymä (E)JES client -sovelluksesta Kelan ympäristössä.

The screenshot displays the EJS2 - EJS client interface with three main windows:

- Job Initiator Status:** A table showing job details.
 

JobID	Status	Classes	JobName	JobID	C	ASID	Owner	System	Row 1 to 11 of 51
1	INACTIVE	A		0009	AS	2678	JES2		
2	ACTIVE	A	TTNSMARG	0087618	A	0002	BRACNCH	AS	2678 JES2
3	INACTIVE	A		00AC	AS	2678	JES2		
4	INACTIVE	A		01AS	AS	2678	JES2		
5	INACTIVE	A		0114	AS	2678	JES2		
6	DRAINED	A		0000	AS	2678	JES2		
7	DRAINED	A		0000	AS	2678	JES2		
8	DRAINED	A		0000	AS	2678	JES2		
9	DRAINED	A		0000	AS	2678	JES2		
10	DRAINED	A		0000	AS	2678	JES2		
11	INACTIVE	F		0007	AS	2678	JES2		
- Job Class Details:** A table showing job class parameters.
 

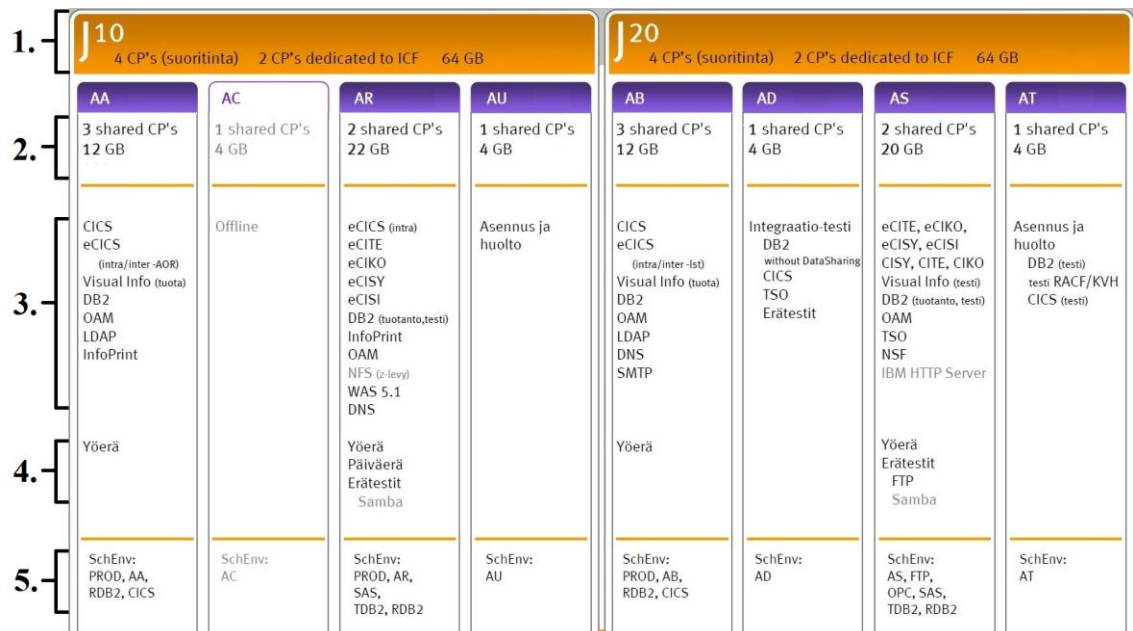
Class	Mbr	Stat	Mode	WaitCt	RunCt	MIStC	MaxRun	MbrAff	MbrCur	MbrMax	ODisp	RLP
A	J67A	JES		0	3	0	0	+	+	+		NO
	J67B											
	J67C											
	J67D											
	J67E											
	J67F											
	J67G											
	J67H											
	J67I											
	J67J											
	J67K											
	J67L											
	J67M											
	J67N											
	J67O											
	J67P											
	J67Q											
	J67R											
	J67S											
	J67T											
	J67U											
	J67V											
	J67W											
	J67X											
	J67Y											
	J67Z											
- Job Operations Log:** A log window showing detailed job execution steps and timestamps, including messages like "PRINTMAY has started to process the document" and "JOB KEY WAS CANCELLED".

Kuvio 21. (E)JES client -sovellus Kelassa.

## 5 KELAN z/OS-JÄRJESTELMÄYMPÄRISTÖ

Kelan suurtietokoneympäristö koostuu kahdesta IBM:n z10 suurtietokoneesta. Koneiden mallityyppi on 2097-E12 ja ne kuuluvat IBM:n Enterprise Class (EC) palvelinluokkaan. Mallityypin loppuosa -E12 tarkoittaa suuren prosessorien maksimilukumäärää, joka siis tässä mallissa on 12 kpl.

Kelan suuren koneiden nimiksi on määritelty J10 ja J20. Molemmat suuren koneet on jaettu neljään LPAR:iin, eli virtuaaliseen järjestelmään. J10 koneella ovat LPAR:it AA, AC, AR ja AU. J20 koneella sijaitsevat LPAR:it AB, AD, AS ja AT. Kriittisiä tuotannollisia palveluita tarjoavia järjestelmiä ovat AA, AB, AR ja AS. Ei-kriittisiä palveluita ja järjestelmän testausta suoritetaan AD-järjestelmässä. AC-järjestelmää pidetään varajärjestelmänä yhteydettömässä tilassa. AT ja AU ovat testausjärjestelmiä, joilla uudet järjestelmät testataan sekä huoltopaketit ajetaan sisään. Ohjelmistojen ja palveluiden osalta LPAR:it muodostavat toisiaan tukevat parit AA-AB, AR-AS, AD-AC ja AT-AU. Jokaisessa LPAR:ssa on käyttöjärjestelmänä vuonna 2008 ilmestynyt z/OS version 1. release 10. Kuviossa 22. on esiteltyä Kelan Parallel Sysplex järjestelmäkokonaisuus, joka koostuu J10 ja J20 suuren koneesta sekä niiden 8 LPAR:sta.



Kuvio 22. Kelan Parallel Sysplex.

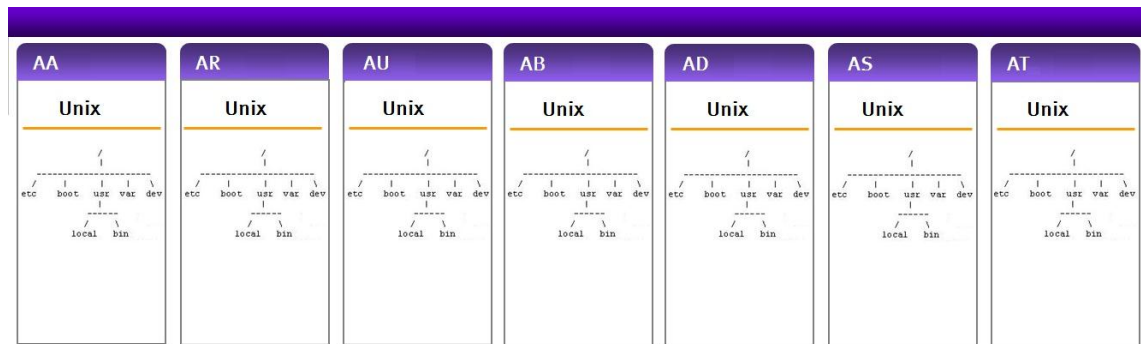
Kuviossa 22. on vasemman reunan numeroinnilla erotettu eri informaatiota käsitteleviä osuuksia. Näiden numeroiden alla käsitellään seuraavat tiedot:

1. Kohdassa yksi määritellään suurkanneen nimi ja sen käyttöön annetut tärkeimmät resurssit. J10 ja J20 koneilla on molemmilla käytettävissään 4 suoritinta, 64 GB muistia sekä kaksi erillistä ICF suoritinta Parallel Sysplex:in sisäistä tiedon-siirtoa varten.
2. Kohdassa kaksi näytetään jokaiselle LPAR:lle määritellyt resurssit, jaetut suorittimet sekä muistin koko.
3. Tässä kohdassa määritellään jokaisen LPAR:in tärkeimmät ohjelmistokokonaisuudet. Tärkeimpiä näistä ovat etuuskäsittelyn tietokantajärjestelmä DB2 ja tapahtumankäsittelyalusta CICS, jonka päälle Kelan suurtietokoneen etuuskäsittely on rakennettu. Verkkopalvelut ottavat yhteyttä etuusjärjestelmiin käyttäen eCICS sekä muita eCI\*\* -alkuisia ohjelmistoja.
4. Tässä määritellään kunkin LPAR:in vastaanottamat eräajoryhmät. Tuotannollisten LPAR:ien alla ajetaan eräajoja. AA ja AB ovat yleensä suurimmalla kuormalla päiväsaikaan, tästä syystä niille sallitaan ainoastaan yöllisiä eräajoja. AR:lle ja AS:lle voivat ohjautua yöeräajojen lisäksi myös muut eräajoryhmät.
5. Kohdassa viisi esitellään jokaiselle LPAR:lle määritellyt SchEnv -muuttujat. SchEnv määritellään eräajolle, tai yksittäiselle ajolle, jonka jälkeen ajo on mahdollista suorittaa missä tahansa LPAR:ssa, jolle on määritelty sama SchEnv. Ajon lähtiessä suoritukseen Parallel Sysplex:in automaattinen kuormanjako huolehtii siitä, että ajo suoritetaan siinä LPAR:ssa, jossa on juuri sillä hetkellä eniten resursseja käytettävissä. Esimerkiksi tietokanta-ajoihin liittyviä, SchEnv=RDB2 -määrittelyksellä olevia ajoja, on mahdollista suorittaa kaikissa tuotannollisissa LPAR:ssa.

Suurin osa MVS-tiedostojärjestelmien tiedostoista on määritelty siten, että ne ovat käytettävissä kaikkialla Parallel Sysplex:ssa. Kriittisten palveluiden käyttämät tiedostot on kaikki määritelty käytettäväksi järjestelmän laajuisesti. Järjestelmäkokonaisuuden laajuinen mahdollisuus käsitellä tiedostoja sallii ohjelmistojen suorittamisen tarvittaessa millä tahansa LPAR:illa sekä ohjelmistojen dynaamisen siirtämisen LPAR:ien välillä. Palveluiden toimintavarmuutta on pyritty parantamaan suorittamalla kriittisiä ohjelmistoja kahdennettuina rinnakkain useammalla LPAR:illa yhtä aikaa.

## 5.1 Jokainen Unix on saari

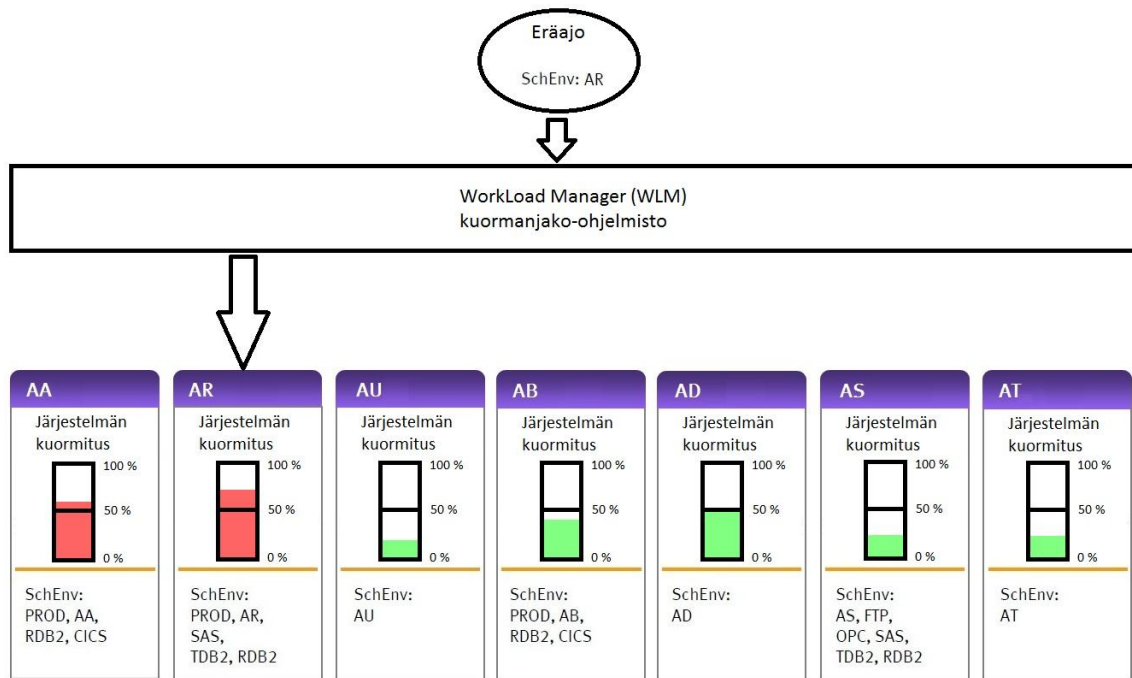
Jokaisella LPAR:illa on oma USS-ympäristönsä, jossa sijaitsee käyttöjärjestelmän sekä siihen integroitujen apuohjelmien käyttämät tiedostoresurssit. Lisäksi käytännössä lähes jokaisella uudehkolla, z/OS käyttöjärjestelmän alla toimivalla ohjelmistotuotteella on tiedostoresursseja Unixin alla. Nykyisessä tilanteessa ongelman muodostaa z/OS-käyttöjärjestelmän rajoittuneisuus liittyen Unix-tiedostoresurssien jakamiseen Parallel Sysplex:in sisällä. Ilman monimutkaista erillistä jakojärjestelmää, eivät yhden LPAR:in omistamat Unix-tiedostoresurssit ole toisten LPAR:ien käytettävissä. Tämä taas on todella haitannut järjestelmän sellaisten sovellusten kahdentamista, jotka toimivat USS-ympäristössä. Kuviossa 23. havainnollistetaan miten jokaisella LPAR:illa on oma rajattu USS-ympäristönsä.



Kuvio 23. Kelan suurkoneissa jokaisella järjestelmällä on eristetty USS-ympäristö.

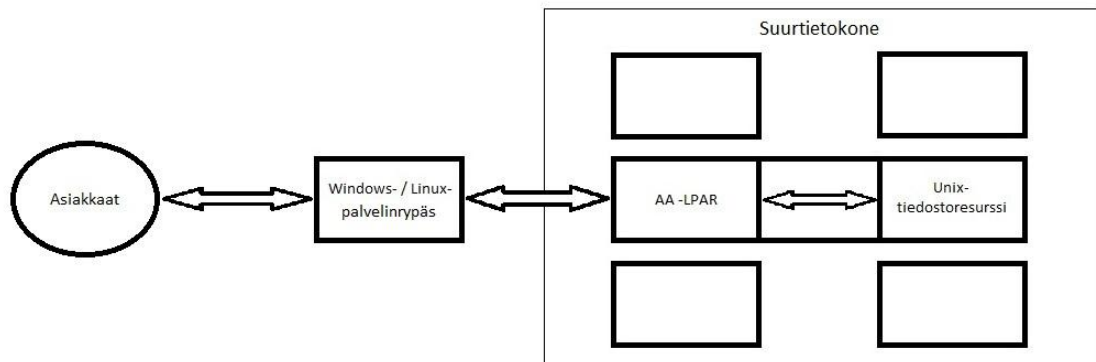
Eristetysti toimivat USS-ympäristöt aiheuttavat sen, että Unix-tiedostoja käyttävien eräajojen suorittaminen täytyy kohdistaa spesifiseen LPAR:iin, koska Unix-tiedostot ovat jokaisessa järjestelmässä uniikkeja. Eräajojen suorittamisen sitominen yhteen tiettyyn LPAR:iin on oikeastaan vastoin Parallel Sysplex:in järjestelmien välisen dynaamisen kuorman jakamisen periaatetta, mutta Unix-tiedostoja käytettäessä tällainen toteutus on pakollista. Kuviossa 24. selvennetään kuinka suurkoneen Unixia käyttävät eräajot joudutaan ohjaamaan tällä hetkellä Kelassa SchEnv -muuttujan avulla tiettyyn yksittäiseen LPAR:iin. Eristetyt USS-ympäristöt aiheuttavat sovelluskehittäjille ylimääräistä työtä, heidän joutuessa ylläpitämään useita, lähes identtisiä Unix-tiedostoympäristöjä samalla ohjelmisto- ja huoltotasolla. Myös levytilaa kuluu hukkaan, kun jokainen USS-ympäristössä käytettävä ohjelmisto täytyy asentaa erikseen sille LPAR:lle, jolla sitä aiotaan käyttää.





Kuvio 24. Uniikit Unix-tiedostot saattavat pakottaa ajamaan eräajoja järjestelmissä, joissa niitä ei kuormituksen näkökulmasta kannattaisi.

Myös palveluiden varmistamisen kannalta pääkoneen Unix-ympäristöjen eristyneisyys tuottaa ongelmia. Palvelut jotka käyttävät toiminnassaan Unixin tiedostoresursseja, eivät voi järkevällä tavalla toimia kahdennettuina usealla LPAR:illa. Tällöin olisi mahdollista tallettaa tietoa useaan toisistaan eristettyyn tiedostojärjestelmään, jolloin tiedon eheys vaarantuisi. Spesifiset USS-ympäristöt pakottavat Kelan ohjelmistot toimimaan siten, että ne ottavat yhteyttä vain yksittäiseen LPAR-ympäristöön, jonka USS-ympäristössä tarvittava sovellus toimii. Yhteen LPAR:iin rajoittava toimintapa on esitelty kuviossa 25.



Kuvio 25. Unix-tiedostoresursseja käytettäessä Kelan sovellusratkaisut ovat sidottuja käyttämään yhtä LPAR:ia.



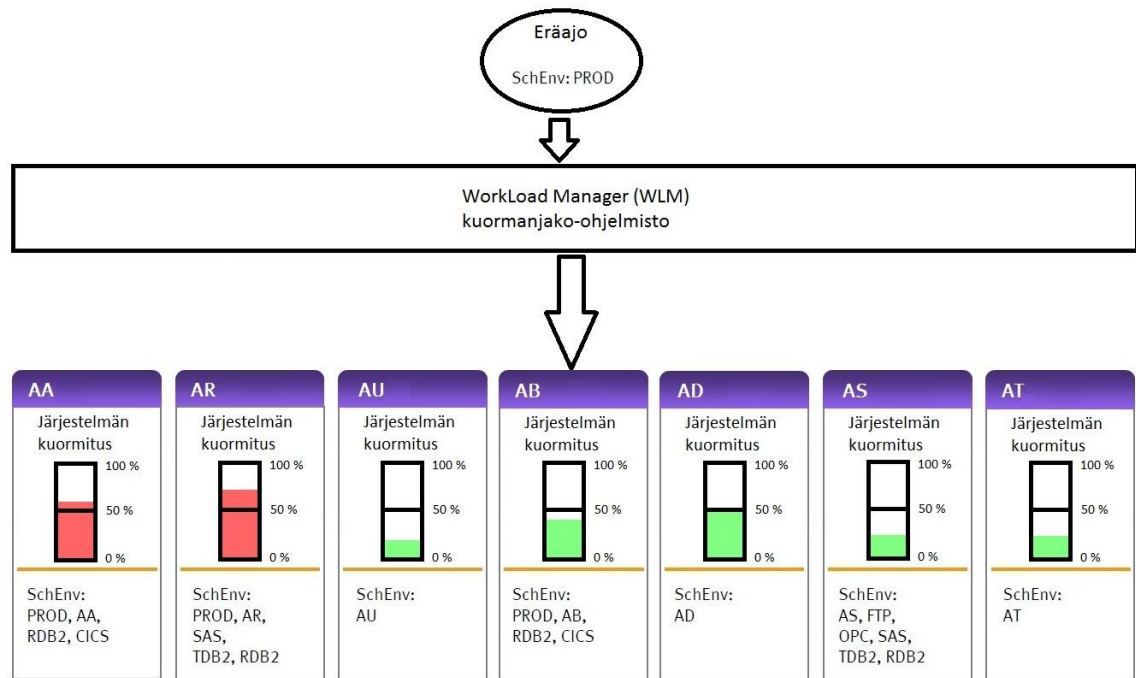
Varsinkin Intel-alustoilla toimivat Kelan verkkopalvelut ottavat usein yhteyden suurtietokoneen etuusjärjestelmään ja tietokantaan USS-ympäristössä toimivien sovellusten kautta. Yksi tärkeimmistä tällaisista sovelluksista USS-ympäristössä on CICS Transaction Gateway (CTG). CTG mahdollistaa Intel-alustoilla sijaitsevien verkkopalveluiden integroinnin pääkoneella sijaitsevaan etuusjärjestelmään. CTG-palvelu on toisin sanoen Kelan verkkopalveluiden kannalta kriittinen ohjelmisto. CTG-palvelun toteutuksesta Kelassa voi lukea enemmän raportista, *WebSphere Application Server:in integrointi CICS-järjestelmään käyttäen CICS Transaction Gateway:ta*, <https://docs.google.com/open?id=0B90FD7Nkp4NAMDA5OTI0MzUtZWY1ZC00NmQ5LTk1MDAtNDViMzUwZTVhMzBl>.

Uudemmissa z/OS-käyttöjärjestelmän versioissa jakamisongelma on korjattu mahdollistamalla zFS-tiedostoresurssien sisällön jakaminen Parallel Sysplex:in LPAR:ien kesken. Tämä huomattava parannus oli suurin yksittäinen syy ryhtyä suunnittelemaan z/OS-käyttöjärjestelmän uusimman version asennusta Kelan ympäristöön.

## 5.2 Jaetun USS-ympäristön mahdollistamat hyödyt ja haasteet

Jaetulla USS-ympäristöllä mahdollistettaisiin kaikkien suurtietokoneen ohjelmistojen ja sovellusten kahdentaminen yhden tai useamman LPAR:in kesken. Ohjelmistojen kahdentaminen taas nostaisi Kelan palveluiden toimintavarmuustasoa ja toisaalta laskisi mahdollisia häiriöaikoja. Tämä olisi todennäköisesti tärkein yksittäinen muutoksen mahdollistama hyöty, sillä Kelan IT-osaston strateginen linjaus on, että kriittiset palvelut eivät saisi olla yksittäisen toiminnan estävän komponentin tai järjestelmän varassa.

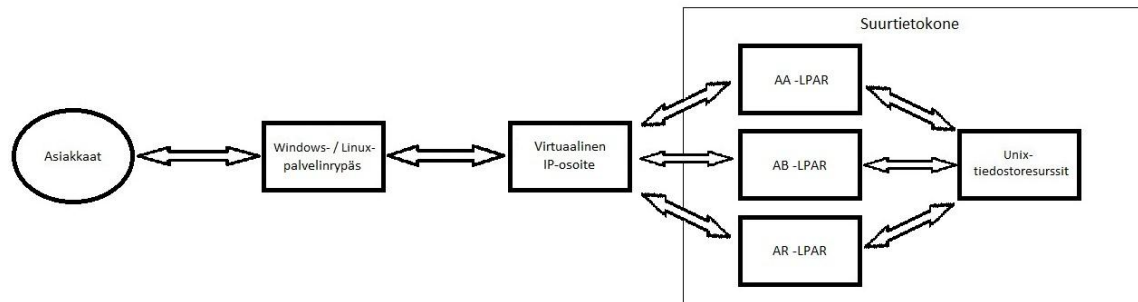
Lisäksi, suuren koneen jaettujen Unix-tiedostojärjestelmien myötä mahdollistetaan Unix-tiedostoja käyttävien eräajojen suorittaminen useammassa kuin yhdessä tiettyssä osiossa. Eräajoja voitaisiin SchEnv -parametrin ja automaattisen kuormanjaon avulla ohjata sellaiseen LPAR:iin, jossa suoritushetkellä on kaikkein matalin kuormitus. Tästä mahdollisuudesta on esimerkki kuviossa 26. Koska Unix-tiedostoresurssit on mahdollista määrittellä tarvittaessa vaikkapa kaikkien LPAR:ien yhteiskäyttöön, voi eräajon määrittellä suoritettavaksi missä tahansa järjestelmiltään muuten sopivassa LPAR:issa.



Kuvio 26. Unix-tiedostot jakamalla eräajot olisi mahdollista suorittaa kuorman kannalta edullisimmassa LPAR:ssa.

Yhteisellä LPAR:ien kesken jaetulla USS-ympäristöllä voitaisiin luopua monikertaisista rinnakkaisista ohjelmistoasennuksista z/OS:n Unix:ssa. MVS-tiedostojärjestelmän alle ohjelmistot on yleensä asennettu yhteiseksi LPAR -parien käyttöön. Tämä tapa voitaisiin siirtää myös USS-ympäristöön, jolloin esimerkiksi LPAR:it AA-AB ja AR-AS voisivat käyttää yhteistä ohjelmistoasennusta. LPAR:ien kesken jaetuilla ohjelmistoilla säästettäisiin levytilaa ja vähennettäisiin rinnakkaisten ohjelmistojen asennuksiin sekä ylläpitämiseen liittyvää työskentelyä. Myös yksikään suurtietokoneen Unix-tiedostoresurssi ei olisi enää pakotettu olemaan yhden ainoan LPAR:in takana, jolloin tiedosto ei ole käytettävissä kyseisen LPAR:in ollessa pois käytöstä.

Suurtietokoneen jaetut Unix-tiedostoresurssit mahdollistaisivat ennen kaikkea Kelan kriittisten ohjelmistojen toimimisen rinnakkain usealla LPAR:illa yhtä aikaa. Tämä lisäisi järjestelmien käytettävyyttä ja vähentäisi oleellisesti palveluiden haavoittuvuutta. Esimerkiksi virtuaalisia IP-osoitteita hyväksi käyttäen voitaisiin Kelan verkkopalveluista tulevia kyselyitä ohjata usean LPAR:in kautta suurtietokoneen etuusarkkitehtuuriin. Tällainen mahdollisuus havainnollistetaan kuviossa 27.



Kuvio 27. Jaetulla USS-ympäristöllä, myös Unix-tiedostoresursseja käyttävien sovel-  
lusten olisi mahdollista toimia rinnakkain useassa LPAR:ssa yhtä aikaa.

Jaettu USS-ympäristö saattaa, varsinkin alkuvaiheessa, aiheuttaa myös yllättäviä haas-  
teita, joita on vaikea nähdä ennalta. Selvää on kuitenkin että suurkanneen  
Unix-tiedostojärjestelmän muokkaamisen suhteen tulee olla erityisen varovainen, sillä  
muutokset eivät vaikuta enää vain yhteen LPAR:iin, vaan käytännössä koko Parallel  
Sysplex:in laajuisesti. Myös hakemistorakenteessa tulee kiinnittää erityistä huomiota  
siihen missä kohdin tehdään muutoksia, sillä kaikkien LPAR:ien ohjelmistot löytyvät  
jostain kohdin yhteisestä hakemistorakenteesta. Tähän asti suurkanneen Unixin alla on  
totuttu siihen, että mahdolliset muutokset vaikuttavat vain siihen järjestelmään johon on  
kirjautunut. Muutoksen jälkeen vaikutukset voivat kuitenkin pahimmillaan olla paljon  
laajemmat. Ohjelmistojen jakamista ajatellessa taas täytyy ottaa huomioon, että virhe  
parametreissa tai huonoksi osoittautuva ohjelmiston päivityspaketti saattaa kaataa oh-  
jelmiston useammalta LPAR:lta. Jossain määrin voitaisiin kaikesti sanoa muutoksien  
tekemisen olevan jaetussa USS-ympäristössä aikaisempaa riskialttiimpaa.

## 6 JAETTUIJEN UNIX-TIEDOSTORESURSSIEN KÄYTTÖÖNOTTAMINEN

Kelassa oli tarkoituksella jätetty väliin z/OS version 1. release 11. käyttöjärjestelmäpainoksen asennus, vaikka kiinnostusta sen mukanaan tuomaan uuteen Unix-resurssien jakamisen mahdollistavaan ominaisuuteen oli ollut. Väliin jättäminen johtui sitä edeltäneen z/OS version 1. release 10. käyttöjärjestelmän pitkäksi venyneestä asennuksesta. Niinpä z/OS version 1. release 12. (z/OS 1.12) käyttöjärjestelmäpainoksen asennuksesta suurkoneympäristön järjestelmiin tehtiin päätös Kelan IT-osastolla jo hyvissä ajoin ennen sen virallista saatavuutta vuoden 2010 syyskuun lopussa. Suurkoneen uusien käyttöjärjestelmäpainosten käyttöönoton suunnittelua helpottaa suuresti Kelan suurkoneen ohjelmistojen lisensiointisopimus, joka suosii uusien versioiden käyttöönottamista. Ylimääräiset kustannukset uusien ohjelmistopainosten asennuksiin liittyen muodostuvat pääasiassa asiantuntijoiden työmääristä.

Käyttöjärjestelmän z/OS 1.12 käyttöönotosta päätettiin perustaa projekti, jonka osaprojektiksi määriteltiin käyttöjärjestelmän asennuksen yhteydessä tapahtuva USS-ympäristöjen yhdistäminen yhdeksi kokonaisuudeksi. Osaprojekti on tämän opinäytetyön kehittämistehtävä, jossa käytännön tasolla toteutetaan tässä työssä kuvatut kehityskohteet Kelan suurkoneympäristöön. Projektin ulkopuolelle rajattiin kaikki jaetun USS-ympäristön mahdollistamat kehityskohteet, kuten esimerkiksi ohjelmistojen kahdentamiset.

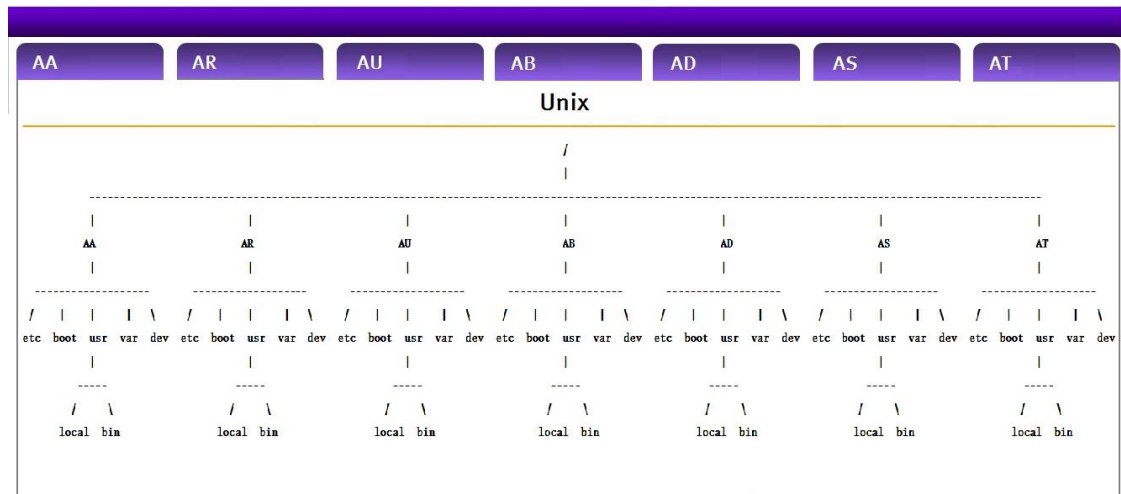
### 6.1 Projekti uuden käyttöjärjestelmän käyttöönotosta

Kelan suurkoneympäristöön liittyvät muutoshankkeet suoritetaan yleensä projektitekniisesti hyvin kevyinä, johtuen ainakin osittain suurkoneasiantuntijoiden vähäiseksi käyneestä lukumäärästä. Projektin suurimmat uhat liittyivät henkilöresursseihin, projektiin liittyvien muutostöiden suorittaminen jäi pääosin kolmen suurkoneasiantuntijan tehtäväksi.

Projektin esiselvitystä suoritettiin pitkin syksyä 2010. Käytössä olevat suurkoneen käyttöjärjestelmät täytyi päivittää asennuspaketeilla, joiden mukana tuli yhteensopivuus z/OS 1.12 käyttöjärjestelmäpainokselle Parallel Sysplex -ympäristöön. Varsinainen projektityö aloitettiin joulukuun alussa, jolloin aloituskokouksessa määriteltiin projektin



tarvitsemat pakolliset hakemistorakenteet ja toisen hakemiston alla muut ohjelmistotuotteen sekä ympäristömuuttujat. Jaetun USS-ympäristön ajatus, jossa jokaisen LPAR:n tiedostoresursseille on määritelty juureen oma hakemistonsa, yksinkertaistetusti esiteltynä kuviossa 29.



Kuvio 29. Suunnitelma jaetun USS-ympäristön juuresta ja sen alle perustettavista LPAR-kohtaisista hakemistoista.

Valtava määrä nykyisiä eräajoja sekä sovelluksia viittaavat johonkin tiettyyn kohtaan Unix-hakemistorakenteessa, josta ne käyttävät tarvitsemaansa tiedostoresurssia. Jaettu USS-ympäristö siirtää vanhat LPAR-kohtaiset Unix-hakemistot uuden yhteisen juuren alle omiin LPAR:in nimen mukaisiin hakemistoihin. Eli esimerkiksi LPAR:in AA nykyinen juuri ja sen alapuolinen hakemistorakenne, käyttöjärjestelmän tiedostoja lukuun ottamatta, olisi käytettävissä uuden jaetun USS-ympäristön juuressa sijaitsevan hakemiston /AA alla. Jaetun USS-ympäristön muutokset hakemistorakenteessa aiheuttavat käytännössä sen, että kaikki nykyisin käytettävät viittaukset hakemistorakenteeseen osoittavat väärin kohtiin. Kaikkien viittausten muuttaminen ei käytännössä ole toteutettavissa projektin puitteissa, joten olemassa olevien viittausten toimivuus päätettiin järjestää linkkitiedostoja käyttämällä. Jaetun USS-ympäristön juureen luodaan linkkitiedostot, jotka käyttävät LPAR:in nimestä muodostettua ympäristömuuttujaa osoittaakseen oikeaan hakemistoon jaetussa ympäristössä. Käytännön esimerkkinä LPAR:illa AS suoritettavan eräajon viittaus /etc -hakemistoon kääntyisi linkkitiedoston avulla /AS/etc -hakemistoon. Näin toteutettuna pakolliset muutokset järjestelmiin säilyvät hallittavalla tasolla jaettuun ympäristöön siirryttäessä, olemassa olevien viittausten toimissa sellaisinaan myös siirron jälkeen.

Hakemistorakenteen suunnittelussa haluttiin yhteinen juuri pitää mahdollisimman yksinkertaisena. Nykyiset toiminnallisuudet mahdollistavien linkkitiedostojen lisäksi juuren alle haluttiin muodostaa ainoastaan pakolliset hakemistot. Tämän jälkeen hakemistorakenteen juuri määriteltäisiin kirjoitussuojatuksi kaikilta käyttäjiltä. Suurkoneen järjestelmänvalvojat voisivat tarvittaessa hetkeksi sallia juureen kohdistuvat muutokset ainoastaan perustelluista syistä. Juureen kiinnitettävä zFS-tiedostoresurssi määriteltiin tarkoituksella pieneksi, koska sinne ei ole tarkoitus tallettaa tavallisia tiedostoja, vaan ainoastaan hakemistoja ja linkkitiedostoja. Juuritasolle määritelyihin hakemistoihin taas kiinnitetään omat kirjoitussuojaamattomat tiedostoresurssinsa, joiden alle on mahdollista juuritasoa vapaammin suunnitella hakemistorakenteita.

Hakemistorakenteen juureen suunnitellut hakemistot ovat:

- LPAR -kohtaiset kahdeksan hakemistoa. Nimetty suoraan LPAR nimen mukaan, hakemistot ovat muotoa /AA, /AB, /AC, /AD, jne.
- Kahdeksan käyttöjärjestelmähakemistoa jokaiselle LPAR:ille. Nimetty siten että toinen ja kolmas merkki kertovat minkä LPAR:in käyttöjärjestelmästä on kysymys, hakemistonimet muodossa /VAA12, /VAB12, /VAC12, /VAD12, jne.
- Käyttäjien kotihakemistojen sijaintihakemisto /u.
- Arkistointijärjestelmän päähakemisto /ars.
- Ohjelmointiympäristöjen päähakemisto /mnt.
- Nykyisen juurihakemistotason varmistus /rootclone.

Järjestelmien yhteisessä USS-ympäristössä korostuu juuritason toiminnan tärkeys. Mikäli juuritaso ei syystä tai toisesta olisi käytettävissä, johtaisi tämä väistämättä koko Parallel Sysplex:ssa Unixin toimimattomuuteen. Tästä syystä juureen kiinnitetty tiedostoresurssi varmistetaan hakemistoon /rootclone kiinnitettyyn tiedostoresurssiin, jonka käyttöjärjestelmä kiinnittää automaattisesti juureen alkuperäisen tiedostoresurssin mahdollisessa vikatilanteessa.

Hakemistorakenteeseen kiinnitettyjen tiedostoresurssien käytettävyyden määrittelyssä on pyritty ottamaan huomioon sen käyttötarkoitus. Määriteltävät zFS-tiedostoresurssit on tarkoitus jakaa käytettäväksi kaikkien LPAR:ien kesken, mutta omistussuhde tiedostoresurssiin määrittelee onko tiedostoresurssi käytettävissä Parallel Sysplex:ssä myös yksittäisten LPAR:ien ollessa pois käytöstä. Mikäli tiedostoresurssin omistaa ainoastaan

yksi LPAR, eivät tiedostoresurssiin määritellyt tiedostot ole käytettävissä omistavan LPAR:in ollessa pois käytöstä. Uuteen hakemistorakenteeseen määritellyt tiedostoresurssit päätettiin määritellä siten, että LPAR -kohtaiset hakemistot sekä käyttöjärjestelmän hakemistot jokaiselle LPAR:lle ovat vain yhden LPAR:in omistuksessa. Nämä tiedostoresurssit sisältävät ainoastaan omistamalleen LPAR:lle tarpeellisia tiedostoja, eikä niiden käytettävyyden varmistaminen ole oleellista muiden LPAR:ien kannalta. Juuren oma tiedostoresurssi ja muut juuren hakemistojen tiedostoresurssit määritellään siten, että ne ovat kaikkien LPAR:ien omistuksessa. Niiden siis pitäisi olla käytettävissä Parallel Sysplex:ssa niin kauan kuin yksikin LPAR on toiminnassa.

Tiedostoresurssien tilaratkaisuissa pyrittiin ottamaan mallia nykyisistä tiedostoresurssista, tarkistaen tilanvarauksia tarvittaessa ylöspäin. Tiedostoresurssit määriteltiin käytettäväksi luku- ja kirjoitusoikeuksin, paitsi juuren tiedostoresurssi, joka määriteltiin ainoastaan lukuoikeuksin. Tarkemmalla tasolla tiedostojen ja hakemistojen oikeudet jaetaan normaalin Unix-käyttöjärjestelmän tapaan.

Unix-tiedostoarkkitehtuurin suunnittelussa pyrittiin varmistamaan nykyisen palvelutason säilyttäminen uudessa USS-ympäristössä siirtämällä tuotannossa olevien ohjelmistojen Unix-hakemistot sellaisinaan LPAR-kohtaisiin hakemistoihin. LPAR:ien kesken jaetut tiedostoresurssit sekä hakemistot kuitenkin mahdollistavat tämän projektin jälkeen palveluiden kehittämisen tavalla, jonka nykyiset rajoitukset ovat estäneet.

### **6.3 Käyttöönottoa edeltävät muutokset**

Ennen z/OS 1.12 -käyttöjärjestelmän asentamisen yhteydessä toteutettavaa USS-ympäristön jakamista täytyi kaikille LPAR:lle tehdä alustavia toimia. Näihin kuuluivat yhteisen Unix-tiedostojärjestelmän käyttämisen Parallel Sysplex ympäristössä mahdollistavien parametrien lisääminen. Sekä suurimpana työnä, tehdä nykyisiä HFS-tiedostoresursseja vastaavat zFS-tiedostoresurssit ja kopioida käytössä olevat tiedot niiden alle.

Kelan nykyinen USS-ympäristö muodostuu pääasiassa HFS-tyyppisistä tiedostoresurssista, jotka on kiinnitetty kunkin LPAR:in Unix-hakemistorakenteeseen alkaen juuritasosta. HFS-tiedostoresurssityypin rinnalla on pitkän aikaa ollut kehittyneempi



zFS-tiedostoresurssityyppi, jonka käyttämistä HFS:n sijasta on IBM:n taholta suositeltu muun muassa sen hiukan nopeamman toiminnan tähden. Kelassa ei kuitenkaan aiemmin ole nähty tarvetta lähteä konvertoimaan käytössä olevia HFS-tiedostoresursseja, vaan ne oli todettu toistaiseksi tarpeeksi toimiviksi ratkaisuiksi Kelan suurkoneympäristössä.

Pakolliset lisättävät tai muutettavat parametrit koostuivat tiedostojärjestelmän jakamisen määrittelystä sekä jokaisen LPAR:in omistukseen tulevien tiedostoresurssien kiinnittämismuuttajista. Muutokset toteutettiin vaiheittain LPAR kerrallaan, aloittaen testausjärjestelmistä ja siirtyen johdonmukaisesti kohti kriittisiä järjestelmiä. Uudet zFS-tiedostoresurssit sekä niiden kiinnittämiseen liittyvät parametrit testattiin rakentamalla uusi hakemistorakenne kokonaisuudessaan nykyisen järjestelmän tilapäishakemiston alle. Testauksella voitiin varmistaa tiedostoresursseista rakentuvan hakemistorakenteen loogisuus, eli jokaiselle hakemistotasolle löytyvät tarvittavat hakemistot joihin zFS-tiedostoresurssit kiinnitetään.

#### **6.4 Käyttöönnoton toteuttaminen**

Jaetun USS-ympäristön käyttöönnotto toteutettiin suurkoneen uuden käyttöjärjestelmän käyttöönnoton yhteydessä keväällä 2011. Käyttöönnotot suoritettiin LPAR kerrallaan aloittaen testausjärjestelmistä ja siirtyen niistä tuotannollisiin järjestelmiin. Yhden LPAR:in käyttöönnottoon varattiin aina yksi sunnuntaipäivä, jolloin kyseinen järjestelmä ohjelmistoinen oli poissa käytöstä. Kriittiset palvelut ohjattiin mahdollisuuksien mukaan käyttämään käyttöönnottojen ajaksi varajärjestelmiä. Käyttöönnottopäiville asetettiin iltoihin takaraja, johon mennessä järjestelmä tulisi saada tarvittavalle toimintatasolle, tai muuten edellinen käyttöjärjestelmäpainos vanhan USS-ympäristön kera palautettaisiin LPAR:lle. Näin varmistettaisiin palveluiden toiminta seuraavalla viikolla, vaikka käyttöönnottoaikatauluja jouduttaisiinkin siirtämään viikolla eteenpäin.

Käyttöönnotto alkoi testausjärjestelmä AU:sta, josta tuli yhteisten zFS-tiedostoresurssien ensimmäinen omistaja. Seuraavaksi käyttöönnotto suoritettiin AT:llä, AD:llä ja AC:llä. Kriittisten järjestelmien AS, AR, AB ja AA käyttöönnotot suoritettiin viimeisinä. LPAR:in käyttöönnoton yhteydessä sen omistamat zFS-tiedostoresurssit kiinnitettiin yhteiseen hakemistorakenteeseen ja niistä tuli osa jaettua USS-ympäristöä. Yhteisomistukseen määritellyille tiedostoresursseille taas käyttöönnotettavasta LPAR:sta tuli yksi sen osa-

omistajista, joka tarkoitti sitä että tiedostoresurssi on jaetussa ympäristössä mukana niin kauan kuin yksikin sen omistavista LPAR:sta on käytettävissä.

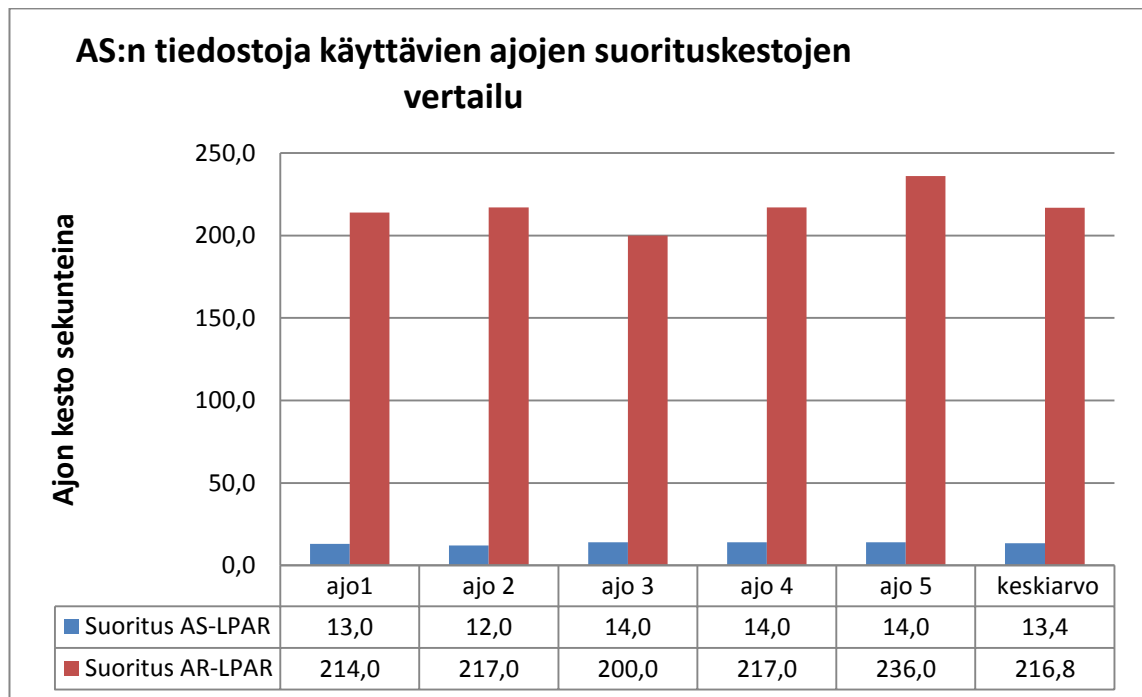
Muutamia pulmia käyttöönottojen yhteydessä kohdattiin, jotka kuitenkin onnistuttiin ratkaisemaan asennuspäivän aikana. Kaikkien LPAR:ien omistukseen tulevat tiedostoresurssit oli muun muassa testattu vain yhden LPAR:in alle tehdyssä hakemistossa. Tästä johtuen ne oli määritelty toimimaan ainoastaan sen yhden LPAR:in alla, eivätkä olleet muiden LPAR:ien käytettävissä. Kaikkien LPAR:ien käyttöönotot onnistuttiin kuitenkin lopulta suorittamaan aikataulujen mukaisesti, eli kohdatut pulmat kyettiin aina ratkaisemaan asennuspäivän kuluessa.

## **6.5 Tiedottaminen, dokumentointi ja mittaaminen**

Jaetun USS-ympäristön suunnitelmat oli laadittu vuonna 2010 Kelan IT-osaston sisällä yhteistyössä arkkitehtuurista ja ohjelmistojen toiminnasta vastaavien henkilöiden sekä järjestelmänvalvojien kesken. Projektin etenemisen aikana tiedottamisesta huolehti projektipäällikkö, joka huolehti osana projektia myös USS-ympäristön muutoksista koskevaa tiedottamista. Kriittisintä tiedottaminen oli käyttöönottopäivien osalta, koska aivan kaikkia palveluita ei ollut mahdollista siirtää LPAR:lta toiselle, vaan tiettyjen LPAR:ien käyttöönottojen osalta oli varauduttava palvelukatkoihin. Tiedottaminen hoidettiin sähköpostin lisäksi Intranet-tiedottamisella ja työvaraussuunnitelmien avulla. Sidosryhmien tiedottaminen ei käyttökatoja lukuun ottamatta ollut projektissa erityisen tärkeässä roolissa, johtuen olemassa olevien palveluiden samanlaisesta toiminnallisuudesta myös jaetussa USS-ympäristössä.

Uuden jaetun USS-ympäristön dokumentointi käsittelee toteutettua muutosta. Suunnitelmiin nähden toteutus pysyi hyvin samanlaisena, kuin mitä ensimmäiset suunnitelmat syksyllä 2010 olivat. Tässä opinnäytetyössä alaotsikon *Unix-tiedostoarkkitehtuurin suunnittelu ja määrittäminen* alla kuvattu muutos toteutettiin ja dokumentoitiin. Jaetun ympäristön hakemistorakenne ja siihen liitetyt tiedostoresurssit kuvattiin Kelan IT-osaston käyttöön. IT-osaston ulkopuolelle ei katsottu tarpeelliseksi levittää käyttöönottoon liittyviä tarkan tason tietoja. Dokumentointi jaetun USS-ympäristön tiedostoresursseista, niiden kiinnityskohdista, tilanvarauksista ja omistavista LPAR:sta löytyy liitteestä 1.

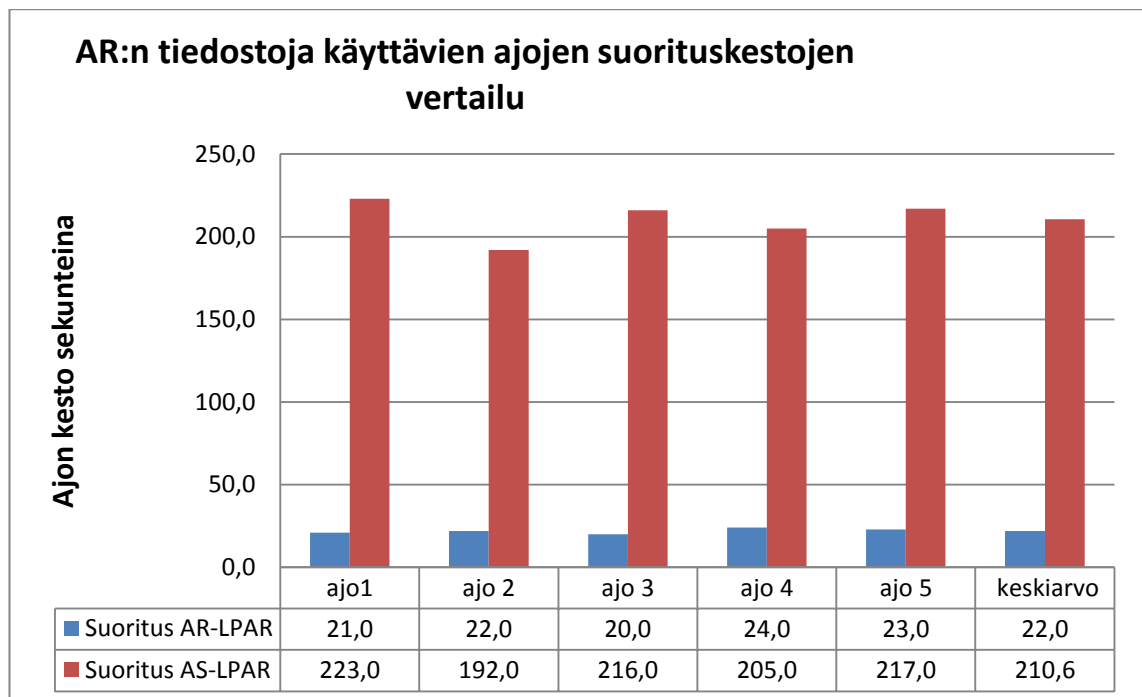
Käyttöönoton jälkeen jaettua USS-ympäristöä testattiin ajoilla, jotka käsittelevät Unixin alla sijaitsevia tiedostoja. Ajoin suoritettiin ensin LPAR:lla, jonka omistukseen käsiteltävät tiedostot kuuluvat ja seuraavaksi sellaisella LPAR:lla, joka ei omista tiedostoja mutta pystyy niitä käyttämään. Suoritettavien ajojen suoritus aika mitattiin, jotta voitaisiin selvittää, minkälainen vaikutus suoritusnopeuteen on sillä omistaako suoritettava LPAR-tiedostoresurssin vai ei. Kuviossa 30. esitetään viiden identtisen testiajon tuloksia, siten että sama ajo on ajettu AS:llä ja sen jälkeen AR:llä. Testiajoin käsittelevät AS-LPAR:in omistamia tiedostoresursseja luku- ja kirjoitusoikeuksin. Testausaikalla AS- ja AR-LPAR:ien kuormituksissa ei ollut mainittavia eroja.



Kuvio 30. Testi Kelan USS-ympäristössä AS-LPAR:in tiedostoresursseja käyttävillä ajoilla millaisia eroja on suorituskestoissa AS- ja AR-LPAR:ien välillä.

Kuten kuvion 30 pylväistä voidaan todeta, oli ajojen suorituskestoissa valtava ero suhteessa siihen suoritettiin ajo AS:llä vai AR:llä. Jokaisessa pylväsparissa vasemmanpuoleinen pylväs kuvastaa ajon suorituskesto AS:n alla ja oikeanpuoleinen pylväs saman ajon suorituskesto AR:n alla toteutettuna. AS-LPAR:in, jonka omistuksessa ajoissa käsiteltävät tiedostot olivat, alla viiden ajon suorituskestojen keskiarvo oli 13,4 sekuntia. AR-LPAR:ssa näiden viiden saman ajon suorituskestojen keskiarvo oli peräti 216,8 sekuntia. Suorituskestoissa nopeusero oli omistavan AS-LPAR:in hyväksi noin 16-kertainen verrattuna AR-LPAR:iin.

Suoritusaikatesti haluttiin suorittaa myös toisin päin, siten että samat viisi ajoa käsitteleviä AR-LPAR:in omistamia tiedostoresursseja luku- ja kirjoitusoikeuksin. Ajo suoritettaisiin yksi kerrallaan ensin AR:llä ja sitten AS:llä. Käsiteltävät tiedostoaineistot eivät ole täysin identtisiä verrattaessa tätä ja edellistä, AS:n tiedostoja käsiteltyä testiä. Se ei kuitenkaan haittaa, koska tarkoitus on verrata tiedostoresurssin omistavan LPAR:in suorituksen tehokkuutta verrattuna LPAR:iin joka ei omista tiedostoresursseja. AR:n omistuksessa olevia tiedostoresursseja käsittelevien ajojen suorituskestojen mittaustuloksia vertaillaan kuviossa 31.



Kuvio 31. Testi Kelan USS-ympäristössä AR-LPAR:in tiedostoresursseja käyttävillä ajoilla millaisia eroja on suorituskestoissa AR- ja AS-LPAR:ien välillä.

Erot LPAR:ien välillä olivat merkittävät myös tässä testissä. Jokaisessa pylvasparissa vasemmanpuoleinen pylvas kuvastaa ajon kestoa AR:ssä ja oikeanpuoleinen AS:ssä. Viiden ajon suorituskestojen keskiarvo AR:n alla oli 22 sekuntia ja AS:n alla 210,6 sekuntia. Tässäkin testissä suorituskestoissa oli suuri ero omistavan LPAR:in hyväksi, AR:n alla ajot suoriutuivat keskimäärin lähes 10-kertaa nopeammin kuin AS:n alla. Huomioon on kuitenkin tärkeää ottaa, että vaikka molemmissa testeissä oli suuret erot LPAR:ien välisissä suorituskestoissa, niin suorittimen (CPU) käytössä ei ollut eroja. Pidempi suoritus aika ei siis lisännyt LPAR:in kuormitusta vaikka suoritus kestitkin huomattavan paljon kauemmin johtuen tiedostojen käytön hitaudesta.

## 7 TULOKSIEN TARKASTELU

Kelan suurtietokoneympäristöön otettiin käyttöön uusi käyttöjärjestelmä, jonka yhteydessä toteutettiin käyttöjärjestelmän Unix-ympäristön tiedostoresurssien jakaminen kaikkien suurkanneen järjestelmien käytettäväksi. Tämän toteutuksen tuloksena syntyi suurkanneen Parallel Sysplex -kokonaisuuden yhteinen Unix-tiedostojärjestelmä, jossa on yhteinen juuri, ja jonka hakemistorakenteen alle kaikki LPAR:it kiinnittävät tiedostoresurssinsa.

### 7.1 Jaettujen resurssien käyttäminen järjestelmien välillä

Yhteinen, jaettu Unix-tiedostojärjestelmä toimii, kuten oli suunniteltukin. Yhteisen juuritason alle on muodostettu hakemistorakenne, johon on kiinnitetty kunkin LPAR:in omat zFS-tiedostoresurssit ja LPAR:ien yhteisomistuksessa olevat zFS-tiedostoresurssit. Hakemistorakenne näyttää kaikista järjestelmistä katsoen identtiseltä.

Tiedostoresurssien sisältöä, hakemistoja ja tiedostoja, on mahdollista käyttää luku- ja kirjoitusoikeuksin kaikista suurkanneen järjestelmistä. Tiedostoresurssit, jotka ovat yhteisomistuksessa, säilyvät kiinnitettyinä ja ovat käytettävissä, vaikka muutamat niiden omistavista LPAR:sta olisikin sammutettuina. Vain yhden LPAR:in omistuksessa olevan tiedostoresurssin kiinnitys taas poistuu hakemistorakenteesta LPAR:in ollessa pois käytöstä, kuten oli tarkoituskin.

Tiedostoresurssin sisällön käyttämiseen vaikuttaa oleellisesti se, mistä LPAR:sta käyttäminen tapahtuu. LPAR, joka omistaa tiedostoresurssin, pystyy käyttämään sitä huomattavan paljon nopeammin kuin LPAR, joka ei omista. Vaikka nopeuseroa oli hieman osattu ennakkoidakin, mittaustuloksissa saadut todella suuret nopeuserot olivat kuitenkin yllätys. Vaikka suorituskestolla ei ollut vaikutusta järjestelmien resurssien kulutukseen, ei suorituskestojen välistä eroa voi olla huomioimatta jatkokehityskohteita suunnitellessa.

## 7.2 Uudet kehitysmahdollisuudet

Suurkoneen LPAR:ien yhteisen Unix-tiedostojärjestelmän avulla mahdollistetaan aiemmin kahdentamattomien ohjelmistojen toimiminen rinnakkain usealla LPAR:lla yhtäaikaaisesti. Yksittäisen, toiminnan kannalta kriittisen, sovelluksen kahdentaminen nostaa palveluiden toimintavarmuutta ja pienentää jatkuvuuden epäonnistumisen riskiä. Ohjelmistojen suunnitelmallinen kahdentaminen jätetään tulevaisuuden kehitysprojektien toteutettavaksi.

Välitön hyöty jaetusta USS-ympäristöstä on tiedostoresurssien saatavuus, kriittisten tiedostoresurssien ei tarvitse enää olla vain yhden LPAR:in takana, vaan tiedostot ovat käytettävissä kaikkialta Parallel Sysplex:sta. Tämä mahdollistaa eräajojen vapaamman ohjaamisen kuormaltaan sopivimmalle LPAR:lle. Eräajojen ohjaamisen suunnittelussa tulee ottaa huomioon LPAR:in tiedostoresurssien omistamisen vaikutus tiedostojen käsittelyn nopeuteen, jotta eräajojen suoritusajat eivät kasva kohtuuttomiksi. Tulevaisuudessa on myös mahdollista säästää levytilaa, luopumalla vielä tällä hetkellä käytössä olevista osittain päällekkäisistä LPAR-kohtaisista hakemistorakenteista, muun muassa käyttöjärjestelmän osalta.

Yleisellä tasolla tunnistettujen riskien vähentämiseen liittyvät kehitysmahdollisuudet, kuten resurssien kahdennus, virheidensietokyvyn nostaminen ja varajärjestelmien perustaminen ovat nyt suurtietokoneympäristössä helpommin toteutettavissa.

## 7.3 Tavoitteiden tarkistus

Unix-tiedostoresurssit ovat käytettävissä koko Kelan suurtietokoneympäristössä. Toteutettu muutos on tehnyt suorkoneympäristön toiminnasta dynaamisempaa, kaikkien Unix-tiedostojen ollessa käytettävissä kaikista järjestelmistä. Palveluiden kahdentamiseen ja sitä kautta riskien pienentämiseen on nyt järjestelmäympäristön kannalta uusia kehitysmahdollisuuksia. Näin ollen tavoitteet voidaan kirjata saavutetuiksi.

## 8 YHTEENVETO

Kelan IT-osasto huolehtii Kelan tietojärjestelmiin liittyvien palveluiden toiminnasta. Monet palveluista ovat Kelan toiminnan kannalta kriittisiä ja tästä syystä niiden jatkuvuuden varmistaminen on ensisijaisen tärkeää. Kelan sosiaaliturvan etuustietojärjestelmät, jotka tuotetaan suurtietokoneympäristössä, ovat palveluista ehkä tärkeimmät. Suurtietokoneympäristöt eroavat suuresti yleisistä Windows- ja Linux-järjestelmistä ja ovat nykyisin käyneet varsin harvinaisiksi. Kelan suurkoneissa on käytössä z/OS-käyttöjärjestelmä. z/OS on monipuolinen ja siksi myös haastavasti hallittava käyttöjärjestelmä, jonka yksi suuri osakokonaisuus on integroitu täysiverinen Unix-käyttöjärjestelmä. z/OS-käyttöjärjestelmässä on siis käytännössä kaksi suurehkoa, toisistaan runsaasti eroavaa käyttöjärjestelmää.

Suurkonejärjestelmissä yhtenä suurimpana haasteena on ollut Unix-tiedostoresurssien jakamisen vaikeus, sen virtuaalisten järjestelmien, LPAR:ien välillä. Kelan IT-osaston palveluiden riskien analysoinnissa tämä on ollut tunnistettu haavoittuvuus, joka on kuitenkin täytynyt hyväksyä hallittuna riskinä. Vuosina 2010 - 2011 Kelan IT-osastolla toteutettiin uuden suurtietokoneen käyttöjärjestelmän asennus, joka mahdollisti Unix-tiedostoresurssien jakamisen toteuttamisen koko suurkoneympäristössä. Tämän opinnäytetyön innoitus kumpuaa tuon uuden käyttöjärjestelmäasennuksen osaprojektina toteutetun Unix-resurssien jakamisen suorittamisesta kohdeympäristöön sekä sen mahdollistamista IT-palveluiden kehittämiskohteista.

Suurkoneympäristön Unix-tiedostoresurssien jakamisen toteutus onnistui suunnitelmien mukaisesti. Tätä kirjoittaessa Parallel Sysplex:in yhteinen Unix-tiedostojärjestelmä on ollut käytössä jo yli vuoden ajan. Käytännön kehittämishankkeet ovat kuitenkin rajoittuneet tähän mennessä eräajojen vapaampaan ohjaamiseen suurkoneympäristöjen välillä. Tämä johtuu vuonna 2011 alkaneesta kilpailuttamisesta, jolla haluttiin ostaa Kelan suurkoneympäristön hoitaminen palveluna ulkopuoliselta toimijalta. Kilpailuttamisen ajaksi ja siihen asti kunnes voittaja ottaisi järjestelmät hoitaakseen, suurkoneen palvelut haluttiin jäädyyttää nykyiselle tasolle, eikä muutoksiin tähtääviä projekteja haluttu aloittaa. Suurkoneen Unix-ympäristöä käyttävien palveluiden kahdentamisen edellytykset ovat kuitenkin tämän työn jälkeen hyvällä tasolla, ja palvelun tarjoajan kanssa yhteistyössä toteutettavissa.

## LÄHTEET

### Kirjalliset lähteet

Jordan, E & Silcock, L. 2005. Strateginen IT-riskien hallinta. Helsinki: Edita

Jaakohuhta, H. 2003. Tietojärjestelmien luotettavuus. Helsinki: Edita

Haikala, I & Järvinen, H-M. 2004. Käyttöjärjestelmät. Jyväskylä: Talentum Media Oy

Crichlow, J. 2001. Hajautetut tietojärjestelmät. Helsinki: Edita

Oksanen, E. 1992. UNIX ja sen hallinta. Hämeenlinna: Karisto Oy

Love, P & Merlino, J & Zimmerman, C & Reed, J & Weinstein, P. 2005. Beginning Unix®. Hoboken: Wiley.

Ceruzzi, P. 2003. a History of modern computing, second edition. Cambridge: The MIT Press.

Rojas, R & Hashagen, U. 2000. The first computers: history and architectures. Cambridge: The MIT Press.

Häggman, K. 1997. Suurten muutosten Suomessa, Kansaneläkelaitos 1937-1997. Jyväskylä: Gummerus.

IBM, z/OS Basic Skills Information Center I. 2010. z/OS concepts.

IBM, z/OS Basic Skills Information Center II. 2010. Application Programming on z/OS.

IBM, z/OS Basic Skills Information Center III. 2008. Mainframe concepts.

IBM, z/OS. 2010. DFSMSdfp Storage Administration.

IBM, z/OS UNIX System Services Planning. 2010. DFSMSdfp Storage Administration.

Valtionhallinnon tietoturvallisuuden johtoryhmä, 2002, VAHTI 1/2002.

Valtionhallinnon tietoturvallisuuden johtoryhmä, 2007, VAHTI 3/2007.

Valtionhallinnon tietoturvallisuuden johtoryhmä, 2010, VAHTI 2/2010.

### Verkkolähteet

z/OS V1R12.0 UNIX System Services Planning. 2010, IBM.

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/BPXZB2B0/](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/BPXZB2B0/)

z/OS V1R12.0 UNIX System Services User's Guide. 2010, IBM.

[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/BPXZA4B0/](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/BPXZA4B0/)



z/OS V1R12.0 UNIX System Services File System Interface Reference, 2010, IBM.  
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/BPXZB5B0/](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/BPXZB5B0/)

z/OS V1R11.0 Distributed File Service zFS Administration. 2009, IBM.  
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/FCXD5A90/](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FCXD5A90/)

z/OS V1R12.0 Planning for Installation. 2010, IBM.  
[http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/E0Z2B1B0/](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/E0Z2B1B0/)

Taloussanomat. 26/01/2010. Luettu 30.08.2012.  
<http://www.taloussanomat.fi/palvelut/2010/01/26/kela-on-internetin-kuningas/20101215/12>

Kelan IT-toiminnan haasteita. 2011. Agora ICT-foorumi.  
[https://www.jyu.fi/erillis/agoracenter/tutkimus/acprojektit/aiitg/foorumit/ict/Markku\\_Suominen\\_Kela\\_150911.pdf](https://www.jyu.fi/erillis/agoracenter/tutkimus/acprojektit/aiitg/foorumit/ict/Markku_Suominen_Kela_150911.pdf)

What is UNIX®? The Open Group. Luettu 11.9.2012.  
[http://www.unix.org/what\\_is\\_unix.html](http://www.unix.org/what_is_unix.html)

The Austin Common Standards Revision Group. Luettu 11.9.2012.  
<http://www.opengroup.org/austin/>

Kelassa hyrräävät tehokkaat koneet. Kauppalehti. Luettu 20.08.2012.  
<http://www.kauppalehti.fi/5/i/talous/uutiset/avoinarkisto/index.jsp?xid=3070667&date=2008/11/25>

# LIIITEET

## Liite 1. USS-ympäristön tiedostoresurssit kiinnityskohtineen 09.09.2011.

1(5)

Juuressa sijaitsevat mountit

```
Filesystem-----512-blocks-----Used--Available--Capacity--Mounted on
NA91.ZFS.ARS.ARS1R      3000960      2430970      569990      82% /ars/r/ars1
NA91.ZFS.ARS.R          18432000     263122      18168878    2% /ars/r
NA91.ZFS.ARS.ARS1S      2355840      137320      2218520     6% /ars/s/ars1
NA91.ZFS.ARS.ARS1K      2355840       61602      2294238     3% /ars/k/ars1
NA91.ZFS.ARS.ARS1T      2355840      98576      2257264     5% /ars/t/ars1
NA91.ZFS.ARS.T          4608000      274198      4333802     6% /ars/t
NA91.ZFS.ARS.S          4608000      47180      4560820     2% /ars/s
NA91.ZFS.ARS.K          4608000      46970      4561030     2% /ars/k
NA91.ZFS.ARS.ARS1G      2355840      697528      1658312    30% /ars/g/ars1
NA91.ZFS.ARS.G          2355840      221520      2134320    10% /ars/g
NA91.ZFS.MNTXCODE       288000       79854      208146     28% /mnt/xcode
NA91.ZFS.MNTSYINT       1440000      14762      1425238     2% /mnt/syint
NA91.PAS1.ZFS1          1440         308         1132       22% /mnt/testimount
NA91.ZFS.MNTRCODE       576000       82742      493258     15% /mnt/rcode
NA91.ZFS.MNTKOULU      1440000      14762      1425238     2% /mnt/koulu
NA91.ZFS.MNTSYSTE      1440000      14762      1425238     2% /mnt/syste
NA91.ZFS.MNTTESTI      1440000      14764      1425236     2% /mnt/testi
NA91.ZFS.MNTTCODE       576000      198812      377188     35% /mnt/tcode
NA91.ZFS.MNTSCODE       288000       79892      208108     28% /mnt/scode
NA91.ZFS.MNTKCODE       288000       79852      208148     28% /mnt/kcode
```

AA mountit

```
Filesystem-----512-blocks-----Used--Available--Capacity--Mounted on
NA91.ZFS.LOG.AA          1152000      283426      868574     25% /AA/var/log
SYS1.ZFS.CFZ.SCFZHFS2.AA2 326880      3444        323436     2% /AA/var/wbem
SYS1.ZFS.SIZUDATA.AA2    221760      2372        219388     2% /AA/var/zosmf/data
NA91.ZFS.VAR.AA          576000       6878        569122     2% /AA/var
NA91.ZFS.TMP.AA          1440000      180872      1259128    13% /AA/tmp
NA91.ZFS.ETC.AA          720000      241190      478810     34% /AA/etc
NA91.ZFS.DEV.AA          72000        1668        70332      3% /AA/dev
NA91.ZFS.AA              5760         316         5444       6% /AA
NA91.HFS.PRINTSRV.AA     4608000      1777336     2783368    39% /AA/var/Printsrv
NA91.ZFS.CM842.AA        14400        2454        11946     18% /VAA12/usr/lpp/icm/V8R4M2
NA91.ZFS.CICSTS41.AA     432000      48930      383070     12% /VAA12/usr/lpp/cicsts/cicsts41
NA91.ZFS.SINGZFS.AA      43200        22296      20904     52% /VAA12/usr/lpp/SA33/ing
NA91.ZFS.TWSV8R5.AA      288000      68194      219806     24% /VAA12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SINGHFS.AA2     34560        22312      12248     65% /VAA12/usr/lpp/ing
SYS1.ZFS.SIBMROOT.AA2    3840         696         3144       19% /VAA12/usr/lpp/pli
SYS1.ZFS.SHPUROOT.AA2    15840        8894        6946       57% /VAA12/usr/lpp/ported
SYS1.ZFS.SHPHROOT.AA2    25920        15858      10062     62% /VAA12/usr/lpp/php
SYS1.ZFS.SHPERROOT.AA2   169920      103314     66606     61% /VAA12/usr/lpp/perl
SYS1.ZFS.SHESROOT.AA2    5760         3604        2156       63% /VAA12/usr/lpp/esa
SYS1.ZFS.SHAPHFS.AA2     69120        11640      57480     17% /VAA12/usr/lpp/IHSA/V7R0
SYS1.ZFS.NETVHFS.AA2     18720        11270      7450       61% /VAA12/usr/lpp/netview/v5r4
SYS1.ZFS.TKGWROOT.AA2    2208         362         1846       17% /VAA12/usr/lpp/kgw
SYS1.ZFS.SBBN7HFS.AA2    5371200     3381710    1989490    63% /VAA12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.SBBNCON1.AA2    908640      9348        899292     2% /VAA12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SEQQROOT.AA2    400320      63710      336610     16% /VAA12/usr/lpp/TWS
SYS1.ZFS.SIZUROOT.AA2    200160      125922     74238     63% /VAA12/usr/lpp/zosmf/V1R12
SYS1.ZFS.TIVOLI.AA2      197280      132616     64664     68% /VAA12/usr/lpp/Tivoli
SYS1.ZFS.XML.AA2         5587200     3486502    2100698    63% /VAA12/usr/lpp/ixm
SYS1.ZFS.JAVA64V6.AA2    1386720     886080     500640     64% /VAA12/usr/lpp/java/J6.0_64
SYS1.ZFS.JAVA31V6.AA2    1239840     792410     447430     64% /VAA12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V5.AA2    817920     521414     296506     64% /VAA12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V5.AA2    864000     550918     313082     64% /VAA12/usr/lpp/java/J5.0
SYS1.ZFS.ROOT.AA2       6298560     4376706    1921854    70% /VAA12
NA91.HFS.CICSTS31.AA     144000      36200      107576     26% /VAA12/usr/lpp/cicsts/cicsts31
NA91.RODEM84.V8R4M0.HFS.AA 432000     351248     80528     82% /VAA12/usr/lpp/ars/V8R4M0
NA91.HFS.RDB2V9.AA       307200      243352     63720     80% /VAA12/usr/lpp/RDB2
NA91.CM84.HFS.AA         172800      169000     3568      98% /VAA12/usr/lpp/icm/V8R4M1
```

AB mountit

```
Filesystem-----512-blocks-----Used--Available--Capacity--Mounted on
NA91.ZFS.LOG.AB          576000      132926     443074     24% /AB/var/log
SYS1.ZFS.CFZ.SCFZHFS2.AB2 326880      3444        323436     2% /AB/var/wbem
SYS1.ZFS.SIZUDATA.AB2    221760      2372        219388     2% /AB/var/zosmf/data
```

2(5)

NA91.ZFS.VAR.AB	144000	1758	142242	2%	/AB/var
NA91.ZFS.TMP.AB	1296000	53900	1242100	5%	/AB/tmp
NA91.ZFS.ETC.AB	720000	362224	357776	51%	/AB/etc
NA91.ZFS.DEV.AB	72000	1668	70332	3%	/AB/dev
NA91.ZFS.AB	5760	316	5444	6%	/AB
NA91.ZFS.CM842.AB	14400	2454	11946	18%	/VAB12/usr/lpp/icm/V8R4M2
NA91.ZFS.SINGZFS.AB	36000	22232	13768	62%	/VAB12/usr/lpp/SA33/ing
NA91.ZFS.CICSTS41.AB	432000	48930	383070	12%	/VAB12/usr/lpp/cicsts/cicsts41
NA91.ZFS.CTG800.AB	144000	81258	62742	57%	/VAB12/usr/lpp/cicstg
NA91.ZFS.TWSV8R5.AB	216000	67458	148542	32%	/VAB12/usr/lpp/TWS/V8R5M0
NA91.ZFS.SINGHFS.AB2	34560	22312	12248	65%	/VAB12/usr/lpp/ing
SYS1.ZFS.SIBMROOT.AB2	3840	696	3144	19%	/VAB12/usr/lpp/pli
SYS1.ZFS.SHPUROOT.AB2	15840	8894	6946	57%	/VAB12/usr/lpp/ported
SYS1.ZFS.SHPHROOT.AB2	25920	15858	10062	62%	/VAB12/usr/lpp/php
SYS1.ZFS.SHPEROOT.AB2	169920	103314	66606	61%	/VAB12/usr/lpp/perl
SYS1.ZFS.SHESROOT.AB2	5760	3604	2156	63%	/VAB12/usr/lpp/esa
SYS1.ZFS.SHAPHFS.AB2	69120	11640	57480	17%	/VAB12/usr/lpp/IHSA/V7R0
SYS1.ZFS.NETVHFS.AB2	18720	11270	7450	61%	/VAB12/usr/lpp/netview/v5r4
NA91.ZFS.TKGWROOT.AB2	2208	362	1846	17%	/VAB12/usr/lpp/kgw
SYS1.ZFS.SBBN7HFS.AB2	5371200	3381710	1989490	63%	/VAB12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.SBBNCON1.AB2	908640	9348	899292	2%	/VAB12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SEQQRROOT.AB2	300960	63694	237266	22%	/VAB12/usr/lpp/TWS
SYS1.ZFS.SIZUROOT.AB2	200160	125922	74238	63%	/VAB12/usr/lpp/zosmf/V1R12
SYS1.ZFS.TIVOLI.AB2	197280	132614	64666	68%	/VAB12/usr/lpp/Tivoli
SYS1.ZFS.XML.AB2	5587200	3486502	2100698	63%	/VAB12/usr/lpp/ixm
SYS1.ZFS.JAVA64V6.AB2	1386720	886080	500640	64%	/VAB12/usr/lpp/java/J6.0_64
SYS1.ZFS.JAVA31V6.AB2	1239840	792410	474730	64%	/VAB12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V5.AB2	817920	521414	296506	64%	/VAB12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V5.AB2	864000	550918	313082	64%	/VAB12/usr/lpp/java/J5.0
SYS1.ZFS.ROOT.AB2	6298560	4376658	1921902	70%	/VAB12
NA91.RODEM84.V8R4M0.HFS.AB	432000	352792	79000	82%	/VAB12/usr/lpp/ars/V8R4M0
NA91.HFS.CICSTS31.AB	144000	36200	107576	26%	/VAB12/usr/lpp/cicsts/cicsts31
NA91.HFS.RDB2V9.AB	307200	243320	63744	80%	/VAB12/usr/lpp/RDB2
NA91.CM84.HFS.AB	172800	169000	3568	98%	/VAB12/usr/lpp/icm/V8R4M1

AD mountit

Filesystem	-----512-blocks-----	Used	Available	Capacity	Mounted on
NA91.ZFS.LOG.AD	720000	21098	698902	3%	/AD/var/log
SYS1.ZFS.CFZ.SCFZHFS2.AD2	326880	57056	269824	18%	/AD/var/wbem
SYS1.ZFS.SIZUDATA.AD2	221760	2372	219388	2%	/AD/var/zosmf/data
NA91.ZFS.VAR.AD	144000	5550	138450	4%	/AD/var
NA91.ZFS.TMP.AD	1000800	146736	854064	15%	/AD/tmp
NA91.ZFS.ETC.AD	144000	94084	49916	66%	/AD/etc
NA91.ZFS.DEV.AD	28800	1236	27564	5%	/AD/dev
NA91.ZFS.AD	5760	316	5444	6%	/AD
SYS1.ZFS.SINGZFS.AD	36000	22232	13768	62%	/VAD12/usr/lpp/SA33/ing
NA91.ZFS.CICSTS41.AD	432000	48930	383070	12%	/VAD12/usr/lpp/cicsts/cicsts41
NA91.ZFS.CTG800.AD	144000	81158	62842	57%	/VAD12/usr/lpp/cicstg
SYS1.ZFS.TWSV8R5.AD	288000	68194	219806	24%	/VAD12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SINGHFS.AD2	34560	22312	12248	65%	/VAD12/usr/lpp/ing
SYS1.ZFS.SIBMROOT.AD2	3840	696	3144	19%	/VAD12/usr/lpp/pli
SYS1.ZFS.SHPUROOT.AD2	15840	8894	6946	57%	/VAD12/usr/lpp/ported
SYS1.ZFS.SHPHROOT.AD2	25920	15858	10062	62%	/VAD12/usr/lpp/php
SYS1.ZFS.SHPEROOT.AD2	169920	103314	66606	61%	/VAD12/usr/lpp/perl
SYS1.ZFS.SHESROOT.AD2	5760	3604	2156	63%	/VAD12/usr/lpp/esa
SYS1.ZFS.SHAPHFS.AD2	69120	43396	25724	63%	/VAD12/usr/lpp/IHSA/V7R0
SYS1.ZFS.NETVHFS.AD2	18720	11270	7450	61%	/VAD12/usr/lpp/netview/v5r4
SYS1.ZFS.TKGWROOT.AD2	2208	362	1846	17%	/VAD12/usr/lpp/kgw
SYS1.ZFS.SBBN7HFS.AD2	5371200	3386878	1984322	64%	/VAD12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.SBBNCON1.AD2	908640	9348	899292	2%	/VAD12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SEQQRROOT.AD2	106560	63646	42914	60%	/VAD12/usr/lpp/TWS
SYS1.ZFS.SIZUROOT.AD2	200160	125922	74238	63%	/VAD12/usr/lpp/zosmf/V1R12
SYS1.ZFS.TIVOLI.AD2	197280	132800	64480	68%	/VAD12/usr/lpp/Tivoli
SYS1.ZFS.XML.AD2	5587200	3486502	2100698	63%	/VAD12/usr/lpp/ixm
SYS1.ZFS.JAVA64V6.AD2	1386720	888784	497936	65%	/VAD12/usr/lpp/java/J6.0_64
SYS1.ZFS.JAVA31V6.AD2	1239840	793946	445894	65%	/VAD12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V5.AD2	817920	523206	294714	64%	/VAD12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V5.AD2	864000	552614	311386	64%	/VAD12/usr/lpp/java/J5.0
SYS1.ZFS.ROOT.AD2	6298560	4376594	1921966	70%	/VAD12
NA91.HFS.DB2V9.AD	460800	243616	217040	53%	/VAD12/usr/lpp/TDB2

AR mountit

Filesystem	-----512-blocks-----	Used	Available	Capacity	Mounted on
NA91.ZFS.CTEK.AR	7800480	6048680	1751800	78%	/AR/ctek
NA91.ZFS.TAKSATRK	410400	28564	381836	7%	/AR/var/smbroot/taksatrk
NA91.ZFS.ATKKTOIM	410400	123504	286896	31%	/AR/var/smbroot/atkktoim

3(5)

NA91.ZFS.TA03OP9.SMB	21600	324	21276	2%	/AR/var/smbroot/ta03op9
NA91.ZFS.THJOHTO.SMB	864000	8900	855100	2%	/AR/var/smbroot/thjohto
NA91.ZFS.ATKJOHTO.SMB	864000	8900	855100	2%	/AR/var/smbroot/atkjohto
NA91.ZFS.VALMIST.SMB	4096800	811812	3284988	20%	/AR/var/smbroot/valmistelu
NA91.ZFS.KUNTOUTU.SMB	864000	8900	855100	2%	/AR/var/smbroot/kuntoutus
NA91.ZFS.TT.SMB	864000	8900	855100	2%	/AR/var/smbroot/tt
NA91.ZFS.ETT.SMB	864000	8900	855100	2%	/AR/var/smbroot/ett
NA91.ZFS.SMB.ROOT	4900320	375332	4524988	8%	/AR/var/smbroot
NA91.ZFS.PRINTSRV.AR	7200000	682848	6517152	10%	/AR/var/Printsrv
NA91.ZFS.LOG.AR	8000640	772788	7227852	10%	/AR/var/log
SYS1.ZFS.CFZ.SCFZHFS2.AR2	326880	57056	269824	18%	/AR/var/wbem
SYS1.ZFS.SIZUDATA.AR2	221760	2372	219388	2%	/AR/var/zosmf/data
NA91.ZFS.VAR.AR	288000	12710	275290	5%	/AR/var
NA91.ZFS.TMP.AR	7200000	5687968	1512032	79%	/AR/tmp
NA91.ZFS.ETC.AR	720000	400116	319884	56%	/AR/etc
NA91.ZFS.DEV.AR	72000	2492	69508	4%	/AR/dev
NA91.ZFS.AR	5760	318	5442	6%	/AR
NA91.HFS.ATKKKM.AR	813120	144480	668480	18%	/AR/var/in/intranet/atkkkm
NA91.HFS.HPUFKI.AR	1094400	1080760	13464	99%	/AR/var/smbroot/hputket
NA91.HFS.JAVABTCH.AR	14169600	8663432	5501624	62%	/AR/var/javabatch
NA91.HFS.VAR.DOMDATA.AR	2836320	31544	2804528	2%	/AR/var/domdata
NA91.HFS.SASDATA.KOHTEET.AR	122976	26976	95840	22%	
/AR/var/sasdata/kelanetti/YHTE/kohteet					
NA91.HFS.SASDATA.TIEDOT.AR	122976	55912	66896	46%	
/AR/var/sasdata/kelanetti/YHTE/tiedot					
NA91.HFS.SASDATA.AR	122976	32816	89920	27%	/AR/var/sasdata
NA91.HFS.FIKEW.LOGARCH	2188800	393264	1795376	18%	/AR/var/logarchive
NA91.HFS.INTERNET.DATA.TUOTA	1003200	264	1002800	1%	/AR/var/in/internet/data/tuota
NA91.HFS.DOCS.AR	2073600	1267328	805296	62%	/AR/var/internet/fikewinteg/docs
NA91.HFS.FIKEW.INTEG.LOG	2188800	224	2188328	1%	/AR/var/internet/fikewinteg/logs
NA91.HFS.FIKEW.INTEG	1632000	192808	1438960	12%	/AR/var/internet/fikewinteg
NA91.ZFS.JAVA14	600480	393250	207230	66%	/VAR12/usr/lpp/java/J1.4
SYS1.ZFS.SINGZFS.AR	36000	22232	13768	62%	/VAR12/usr/lpp/SA33/ing
NA91.ZFS.CM842.AR	14400	2454	11946	18%	/VAR12/usr/lpp/icm/V8R4M2
NA91.WEBSTU51.AR.ZFS	7200	374	6826	6%	/VAR12/usr/lpp/dmh
NA91.ZFS.CICSTS41.AR	432000	48930	383070	12%	/VAR12/usr/lpp/cicsts/cicsts41
NA91.ZFS.CTG800.AR	144000	81274	62726	57%	/VAR12/usr/lpp/cicstg
SYS1.ZFS.TWSV8R5.AR	288000	68194	219806	24%	/VAR12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SINGHFS.AR2	34560	22312	12248	65%	/VAR12/usr/lpp/ing
SYS1.ZFS.SIBMROOT.AR2	3840	696	3144	19%	/VAR12/usr/lpp/pli
SYS1.ZFS.SHPURROOT.AR2	15840	8894	6946	57%	/VAR12/usr/lpp/ported
SYS1.ZFS.SHPHROOT.AR2	25920	15858	10062	62%	/VAR12/usr/lpp/php
SYS1.ZFS.SHPERROOT.AR2	169920	103314	66606	61%	/VAR12/usr/lpp/perl
SYS1.ZFS.SHESROOT.AR2	5760	3604	2156	63%	/VAR12/usr/lpp/esa
SYS1.ZFS.SHAPHFS.AR2	69120	43396	25724	63%	/VAR12/usr/lpp/IHSA/V7R0
SYS1.ZFS.NETVHFS.AR2	18720	11270	7450	61%	/VAR12/usr/lpp/netview/v5r4
SYS1.ZFS.TKGWROOT.AR2	2208	362	1846	17%	/VAR12/usr/lpp/kgw
SYS1.ZFS.SBBN7HFS.AR2	5371200	3386878	1984322	64%	/VAR12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.SBBNCON1.AR2	908640	9348	899292	2%	/VAR12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SEQQROOT.AR2	106560	63646	42914	60%	/VAR12/usr/lpp/TWS
SYS1.ZFS.SIZUROOT.AR2	200160	125922	74238	63%	/VAR12/usr/lpp/zosmf/V1R12
SYS1.ZFS.TIVOLI.AR2	197280	133100	64180	68%	/VAR12/usr/lpp/Tivoli
SYS1.ZFS.XML.AR2	5587200	3486502	2100698	63%	/VAR12/usr/lpp/ixm
SYS1.ZFS.JAVA64V6.AR2	1386720	888784	497936	65%	/VAR12/usr/lpp/java/J6.0_64
SYS1.ZFS.JAVA31V6.AR2	1239840	793930	445910	65%	/VAR12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V5.AR2	817920	523206	294714	64%	/VAR12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V5.AR2	864000	552630	311370	64%	/VAR12/usr/lpp/java/J5.0
SYS1.ZFS.ROOT.AR2	6298560	4377342	1921218	70%	/VAR12
NA91.HFS.CICSTS31.AR	144000	36200	107576	26%	/VAR12/usr/lpp/cicsts/cicsts31
NA91.TONDEM84.V8R4M0.HFS.AR	432000	352120	79664	82%	/VAR12/usr/lpp/ars/V8R4M0
NA91.HFS.USRSPOOL.AR	960000	32296	927480	4%	/VAR12/usr/spool
NA91.HFS.RDB2V9.AR	307200	107456	199512	36%	/VAR12/usr/lpp/RDB2
NA91.HFS.DB2V9.AR	460800	243584	217080	53%	/VAR12/usr/lpp/TDB2

AS mountit

Filesystem	-----512-blocks-----	Used	Available	Capacity	Mounted on
NA91.ZFS.SMPEHPZ.AS	2880000	2101678	778322	73%	/AS/na03/shopz
NA91.ZFS.SMPEDB2.AS	7200000	1960298	5239702	28%	/AS/na03/smpe/db2
NA91.ZFS.SMPEZOS.AS	14400000	4901782	9498218	35%	/AS/na03/smpe/zos
NA91.ZFS.LOG.AS	1296000	1262432	33568	98%	/AS/var/log
SYS1.ZFS.CFZ.SCFZHFS2.AS2	326880	57056	269824	18%	/AS/var/wbem
SYS1.ZFS.SIZUDATA.AS2	221760	2372	219388	2%	/AS/var/zosmf/data
NA91.ZFS.VAR.AS	1000800	43574	957226	5%	/AS/var
NA91.ZFS.TMP.AS	12960000	11309244	1650756	88%	/AS/tmp
NA91.ZFS.ETC.AS	2880000	150592	2729408	6%	/AS/etc
NA91.ZFS.DEV.AS	72000	1668	70332	3%	/AS/dev
NA91.ZFS.AS	5760	332	5428	6%	/AS

4(5)

NA91.HFS.SASDATA.AS	122976	8640	114096	8%	/AS/var/sasdata
NA91.HFS.SASSTP.AS	20544	48	20360	1%	/AS/var/stp
NA91.HFS.VAR.MAN.AS	4608	3360	1008	77%	/AS/var/man
NA91.HFS.WWWPLSRV.LOG.AS	147264	142888	4104	98%	/AS/var/ibmhttp/wwwplsrv/log
NA91.HFS.WWWPLSRV.PUB.AS	96	40	0	100%	/AS/var/ibmhttp/wwwplsrv/pub
NA91.HFS.WWWPLSRV.AS	2688	536	1904	22%	/AS/var/ibmhttp/wwwplsrv
NA91.HFS.KELASTO.AS	2377920	2176400	201296	92%	/AS/kelanelt/kelasto
NA91.HFS.BARONET.AS	870528	866112	3768	100%	/AS/kelanelt/baronet
NA91.HFS.INTRANET.AS	5475456	4587296	884000	84%	/AS/kelanelt
NA91.HFS.KELAPORT.AS	223104	160	222568	1%	/AS/kelaportal
NA91.ZFS.CM842.AS	14400	2454	11946	18%	/VAS12/usr/lpp/icm/V8R4M2
NA91.ZFS.SAS92.AS	36000000	28359626	7640374	79%	/VAS12/usr/lpp/SAS92/Depot
NA91.ZFS.SAS92BAS.AS	8064000	3305060	4758940	41%	/VAS12/usr/lpp/SAS92/SAS
NA91.ZFS.SAS92CFG.AS	8064000	983668	7080332	13%	/VAS12/usr/lpp/SAS92/Config
NA91.ZFS.SMB.BOOKSRV.AS	10241280	3794948	6446332	38%	/VAS12/usr/lpp/booksrv/pdfbooks
NA91.ZFS.OMEGADB2.AS	28800	12038	16762	42%	/VAS12/usr/lpp/opmei
NA91.ZFS.SINGZFS.AS	70560	24660	45900	35%	/VAS12/usr/lpp/SA33/ing
NA91.ZFS.CICSTS41.AS	432000	48932	383068	12%	/VAS12/usr/lpp/cicsts/cicsts41
NA91.ZFS.CTG800.AS	400320	81412	318908	21%	/VAS12/usr/lpp/cicstg
NA91.ZFS.TWSV8R5.AS	216000	67458	148542	32%	/VAS12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SINGHFS.AS2	34560	22312	12248	65%	/VAS12/usr/lpp/ing
SYS1.ZFS.SIBMROOT.AS2	3840	696	3144	19%	/VAS12/usr/lpp/pli
SYS1.ZFS.SHPUROOT.AS2	15840	8894	6946	57%	/VAS12/usr/lpp/ported
SYS1.ZFS.SHPHROOT.AS2	25920	15858	10062	62%	/VAS12/usr/lpp/php
SYS1.ZFS.SHPEROOT.AS2	169920	103314	66606	61%	/VAS12/usr/lpp/perl
SYS1.ZFS.SHESROOT.AS2	5760	3604	2156	63%	/VAS12/usr/lpp/esa
SYS1.ZFS.SHAPHFS.AS2	69120	43396	25724	63%	/VAS12/usr/lpp/IHSA/V7R0
SYS1.ZFS.NETVHFS.AS2	18720	11270	7450	61%	/VAS12/usr/lpp/netview/v5r4
SYS1.ZFS.TKGWROOT.AS2	2208	362	1846	17%	/VAS12/usr/lpp/kgw
SYS1.ZFS.SBBN7HFS.AS2	5371200	3386878	1984322	64%	/VAS12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.SBBNCON1.AS2	908640	9348	899292	2%	/VAS12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SEQQROOT.AS2	106560	63646	42914	60%	/VAS12/usr/lpp/TWS
SYS1.ZFS.SIZUROOT.AS2	200160	125922	74238	63%	/VAS12/usr/lpp/zosmf/V1R12
SYS1.ZFS.TIVOLI.AS2	197280	134460	62820	69%	/VAS12/usr/lpp/Tivoli
SYS1.ZFS.XML.AS2	5587200	3486502	2100698	63%	/VAS12/usr/lpp/ixm
SYS1.ZFS.JAVA64V6.AS2	1386720	888784	497936	65%	/VAS12/usr/lpp/java/J6.0_64
SYS1.ZFS.JAVA31V6.AS2	1239840	793930	445910	65%	/VAS12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V5.AS2	817920	523206	294714	64%	/VAS12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V5.AS2	864000	864000	311370	64%	/VAS12/usr/lpp/java/J5.0
SYS1.ZFS.ROOT.AS2	8000640	6130372	1870268	77%	/VAS12
NA91.HFS.CICSTS31.AS	144000	36208	107568	26%	/VAS12/usr/lpp/cicsts/cicsts31
NA91.HFS.SAS9LV2.AS	362880	276992	85704	77%	/VAS12/usr/lpp/SAS9
NA91.HFS.RDZ750.AS	41184	38760	2184	95%	/VAS12/usr/lpp/rdz
NA91.TONDEM84.V8R4M0.HFS.AS	432000	349104	82680	81%	/VAS12/usr/lpp/ars/V8R4M0
NA91.HFS.USRSPPOOL.AS	475488	1576	473672	1%	/VAS12/usr/spool
NA91.CM84.HFS.AS	511200	506920	4064	100%	/VAS12/usr/lpp/icm/V8R4M1
NA91.HFS.RDB2V9.AS	307200	43160	263800	15%	/VAS12/usr/lpp/RDB2
NA91.HFS.DB2V9.AS	460800	243664	217008	53%	/VAS12/usr/lpp/TDB2
NA91.HFS.WEBSTU51.AS	729792	266816	462736	37%	/VAS12/usr/lpp/dmh

AT mountit

Filesystem	-----512-blocks-----	Used	Available	Capacity	Mounted on
NA91.ZFS.AT	5760	316	5444	6%	/AT
NA91.ZFS.DEV.AT	28800	1236	27564	5%	/AT/dev
NA91.ZFS.ETC.AT	734400	382656	351744	53%	/AT/etc
NA91.ZFS.TMP.AT	8000640	1180814	6819826	15%	/AT/tmp
NA91.ZFS.VAR.AT	266400	100240	166160	38%	/AT/var
SYS1.ZFS.SIZUDATA.AT	221760	2372	219388	2%	/AT/var/zosmf/data
SYS1.ZFS.CFZ.SCFZHFS2.AT	326880	57056	269824	18%	/AT/var/wbem
NA91.ZFS.LOG.AT	1152000	947676	204324	83%	/AT/var/log
NA91.HFS.VAR.SASDATA.AT	1012320	11392	1000680	2%	/AT/var/sasdata
SYS1.ZFS.ROOT.AT	6298560	4380284	1918276	70%	/VAT12
SYS1.ZFS.JAVA31V5.AT	864000	552614	311386	64%	/VAT12/usr/lpp/java/J5.0
SYS1.ZFS.JAVA64V5.AT	817920	523206	294714	64%	/VAT12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V6.AT	1239840	793914	445926	65%	/VAT12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V6.AT	1386720	888784	497936	65%	/VAT12/usr/lpp/java/J6.0_64
SYS1.ZFS.XML.AT	5587200	3486502	2100698	63%	/VAT12/usr/lpp/ixm
SYS1.ZFS.TIVOLI.AT	197280	132838	64442	68%	/VAT12/usr/lpp/Tivoli
SYS1.ZFS.SIZUROOT.AT	200160	125922	74238	63%	/VAT12/usr/lpp/zosmf/V1R12
SYS1.ZFS.SEQQROOT.AT	106560	63646	42914	60%	/VAT12/usr/lpp/TWS
SYS1.ZFS.SBBNCON1.AT	908640	9348	899292	2%	/VAT12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SBBN7HFS.AT	5371200	3386878	1984322	64%	/VAT12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.TKGWROOT.AT	2208	362	1846	17%	/VAT12/usr/lpp/kgw
SYS1.ZFS.NETVHFS.AT	18720	11270	7450	61%	/VAT12/usr/lpp/netview/v5r4
NA91.ZFS.TWSV8R5.AT	360000	68930	291070	20%	/VAT12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SHAPHFS.AT	69120	11992	57128	18%	/VAT12/usr/lpp/IHSA/V7R0
SYS1.ZFS.SHESROOT.AT	5760	3604	2156	63%	/VAT12/usr/lpp/esa

5(5)

SYS1.ZFS.SHPEROOT.AT	169920	103314	66606	61%	/VAT12/usr/lpp/perl
SYS1.ZFS.SHPHROOT.AT	25920	15858	10062	62%	/VAT12/usr/lpp/php
SYS1.ZFS.SHPUROOT.AT	15840	8894	6946	57%	/VAT12/usr/lpp/ported
SYS1.ZFS.SIBMROOT.AT	3840	696	3144	19%	/VAT12/usr/lpp/pli
SYS1.ZFS.SINGHFS.AT	34560	22312	12248	65%	/VAT12/usr/lpp/ing
NA91.ZFS.SINGZFS.AT	36000	22232	13768	62%	/VAT12/usr/lpp/SA33/ing
NA91.ZFS.CTG800.AT	504000	84832	419168	17%	/VAT12/usr/lpp/cicstg
NA91.ZFS.CICSTS41.AT	432000	48978	383022	12%	/VAT12/usr/lpp/cicsts/cicsts41
NA91.HFS.USR.LPP.CICSTS.CICSTS31.AT	72000		36144	35632	51%
/VAT12/usr/lpp/cicsts/cicsts31					
NA91.UONDEM84.V8R4M0.HFS.AT	432000	349152	82632	81%	/VAT12/usr/lpp/ars/V8R4M0
NA91.HFS.WEBSTU51.AT	729792	265080	464472	37%	/VAT12/usr/lpp/dmh
NA91.HFS.AUD2V9.AT	82272	43456	38576	53%	/VAT12/usr/lpp/AUD2
NA91.HFS.DB2V9.AT	72000	43416	28344	61%	/VAT12/usr/lpp/TDB2

AU mountit

Filesystem	-----512-blocks-----	Used	Available	Capacity	Mounted on
NA91.ZFS.AU	5760	4460	1300	78%	/AU
NA91.ZFS.DEV.AU	72000	1668	70332	3%	/AU/dev
NA91.ZFS.ETC.AU	1036800	86366	950434	9%	/AU/etc
NA91.ZFS.TMP.AU	1440000	269426	1170574	19%	/AU/tmp
NA91.ZFS.VAR.AU	1036800	194934	841866	19%	/AU/var
SYS1.ZFS.SIZUDATA	221760	2372	219388	2%	/AU/var/zosmf/data
SYS1.ZFS.CFZ.SCFZHFS2	326880	104286	222594	32%	/AU/var/wbem
NA91.ZFS.SMPEZOS.AU	8380800	184524	8196276	3%	/AU/na03/smpe/zos
NA91.ZFS.SMPEDB2.AU	7200000	87368	7112632	2%	/AU/na03/smpe/db2
SYS1.HFS.LOG	15408000	14061264	1346456	92%	/AU/var/log
NA91.HFS.PRINTSRV.AU	4608000	318184	4289616	7%	/AU/var/Printsrv
SYS1.ZFS.ROOT	6298560	4373902	1924658	70%	/VAU12
SYS1.ZFS.JAVA31V5	864000	554008	309992	65%	/VAU12/usr/lpp/java/J5.0
SYS1.ZFS.JAVA64V5	817920	524456	293464	65%	/VAU12/usr/lpp/java/J5.0_64
SYS1.ZFS.JAVA31V6	1239840	802622	437218	65%	/VAU12/usr/lpp/java/J6.0
SYS1.ZFS.JAVA64V6	1386720	900770	485950	65%	/VAU12/usr/lpp/java/J6.0_64
SYS1.ZFS.XML	5587200	3484678	2102522	63%	/VAU12/usr/lpp/ixm
SYS1.ZFS.TIVOLI	197280	133328	63952	68%	/VAU12/usr/lpp/Tivoli
SYS1.ZFS.SIZUROOT	200160	125940	74220	63%	/VAU12/usr/lpp/zosmf/V1R12
SYS1.ZFS.SEQOROOT	106560	63646	42914	60%	/VAU12/usr/lpp/TWS
SYS1.ZFS.SBBNCON1	908640	9348	899292	2%	/VAU12/zWebSphereOEM/V7R0/config1
SYS1.ZFS.SBBN7HFS	5371200	3093938	2277262	58%	/VAU12/usr/lpp/zWebSphereOEM/V7R0
SYS1.ZFS.TKGWROOT	2208	362	1846	17%	/VAU12/usr/lpp/kgw
SYS1.ZFS.NETVHFS	18720	11270	7450	61%	/VAU12/usr/lpp/netview/v5r4
SYS1.ZFS.TWSV8R5.AU	360000	68930	291070	20%	/VAU12/usr/lpp/TWS/V8R5M0
SYS1.ZFS.SHAPHFS	69120	43528	25592	63%	/VAU12/usr/lpp/IHSA/V7R0
SYS1.ZFS.SHESROOT	5760	3604	2156	63%	/VAU12/usr/lpp/esa
SYS1.ZFS.SHPEROOT	169920	103314	66606	61%	/VAU12/usr/lpp/perl
SYS1.ZFS.SHPHROOT	25920	15858	10062	62%	/VAU12/usr/lpp/php
SYS1.ZFS.SHPUROOT	15840	8894	6946	57%	/VAU12/usr/lpp/ported
SYS1.ZFS.SIBMROOT	3840	696	3144	19%	/VAU12/usr/lpp/pli
SYS1.ZFS.SINGHFS	34560	22312	12248	65%	/VAU12/usr/lpp/ing
SYS1.ZFS.SINGZFS.AU	43200	22296	20904	52%	/VAU12/usr/lpp/SA33/ing
NA91.ZFS.DB2V9.AU	432000	264162	167838	62%	/VAU12/usr/lpp/AUD2
NA91.UONDEM84.V8R4M0.HFS	432000	350688	81096	82%	/VAU12/usr/lpp/ars/V8R4M0