

Sveučilište u Zagrebu
PMF – Matematički odjel

Jurko Gospodnetić

Automatizirana mrežna
instalacija operacijskih sustava
(Fully Automatic Installation)

Diplomski rad

Zagreb, studenog 2002.

Sadržaj

1.	Uvod.....	3
2.	Operacijski sustav.....	5
2.1.	Kratka povijest UNIX i Linux operacijskih sustava	5
2.2.	Veza UNIX i Linux operacijskih sustava.....	5
2.3.	Linux distribucije	6
2.4.	Debian GNU/Linux distribucija.....	7
3.	Standardna instalacija operacijskog sustava	9
3.1.	Tradicionalna instalacija	9
3.2.	Mrežna instalacija.....	9
3.3.	Debian GNU/Linux instalacija	10
3.4.	Automatizirana mrežna instalacija.....	11
4.	Opis općeg sustava za automatizirano centralno održavanje mreže računala	13
4.1.	Osnovni dijelovi sustava	13
4.1.1.	Gold server	13
4.1.2.	Install serveri	13
4.1.3.	Klijenti	14
4.2.	Održavanje kontrole nad sustavom	15
5.	FAI sustav.....	17
5.1.	O FAI sustavu.....	17
5.2.	Metoda razvoja FAI sustava – open source.....	17
5.3.	Posebnosti korištene implementacije FAI sustava	17
5.4.	Prvi pogled na rad FAI sustava	18
5.4.1.	Osnovne postavke FAI servera	18
5.4.2.	Podizanje FAI klijenta preko mreže.....	19
5.4.3.	Spašavanje klijenata.....	19
5.4.4.	Neke mogućnosti FAI sustava	20
5.5.	Osnovni dijelovi FAI sustava	20
5.5.1.	FAI server.....	20
5.5.2.	Mrežni operacijski sustav	22
5.5.3.	Povezanost FAI servera i klijenata	23
5.6.	Instalacija FAI servera	24
5.6.1.	Priprema za instalaciju	24
5.6.2.	Instalacija	24
5.7.	Konfiguracija mreže FAI klijenata	28
5.7.1.	Osnovna konfiguracija mreže klijenata - DHCP.....	28
5.7.2.	Konfiguracija FAI automatske mrežne instalacije	32
5.8.	Daljnji razvoj FAI sustava	35
6.	Zaključak	37
7.	Dodaci	39
7.1.	Mrežno podizanje Debian GNU/Linux operacijskog sustava	39
7.1.1.	Bootstrap kod	39
7.1.2.	Opis postupka podizanja mrežnog operacijskog sustava	39
7.2.	Konfiguriranje obrade klijentovih lokalnih diskova.	41
7.2.1.	Format konfiguracijskih datoteka	41
7.3.	Neke korisne WWW stranice.....	44
8.	Literatura	45

1. Uvod

Kod postavljanja i administriranja infrastrukture jednog računalnog sustava, uvriježeno je razmišljati u terminima pojedinih računala umjesto gledanja na cijelu infrastrukturu kao na cjelinu. Ta uvriježena praksa stvara mnoge probleme, uključujući aministraciju koja iziskuje velike količine aktivnog rada, veliku cijenu cijele infrastrukture, ograničen pristup infrastrukturi bilo kakvom kontrolnom mehanizmu te općenito malo dostupnih izvora znanja i koda iskoristivog za administriranje velikih infrastrukture.

Ovdje opisani model tretira infrastrukturu kao jedan veliki distribuirani virtualni stroj te nam omogućuje puno efikasniji pristup cijelom problemu velikih računalnih infrastrukture. Ovakav model ispitan je i razvijen tokom višegodišnjeg rada na infrastrukturama veličine od nekoliko pa do 300-1000 čvorova te se pokazalo da su isti principi primjenjivi kako na male tako i na velike sustave. Primjeri mjesta gdje se ovom metodom održavaju infrastrukture su NASA, mnoga velika sveučilišta diljem svijeta te velike svjetske burze.

Prethodnih radova koji se bave problematikom velikih računalnih infrastrukture u bilo kakvom općem smislu relativno je malo u tisku. Zahvaljujući radu mnoštva predanih ljudi, danas imamo veliku pokrivenost pojedinih alata, tehnika i pristupa, ali teško je u praksi naći kako to sve objединiti zajedno u jednu cjelinu koja se bavi problematikom skupine više od desetak računala.

Ovdje je pokušano pristupiti općenito tom problemu te prikazati jedno njegovo rješenje koje nije u svom začetku ograničeno te može skalirati do primjene na velikim računalnim infrastrukturama. Posebnu pažnju ovdje posvećujemo postupku instalacije operacijskih sustava na toj računalnoj infrastrukturi koji za razliku od tradicionalnog omogućuje lako, brzo, jednostavno i jeftinije održavanje cijelog sustava.

Osnova cijelog pristupa je gledanje na infrastrukturu kao jednu cjelinu umjesto kao nakupinu pojedinih dijelova. Ta promjena gledišta, kao i odluke koje ona izaziva, čini ogromnu razliku u cijeni te lakoći održavanja i administriranja cijelog sustava.

Cijela metodologija opisana je na Linux operacijskom sustavu, točnije na Debian GNU/Linux distribuciji tog operacijskog sustava, ali isti principi su potpuno primjenjivi te ne manje potrebni, u infrastrukturama osnovanim na drugim operacijskim sustavima.

Svo istraživanje i razvoj koji je prethodio ovom radu napravljen je u sklopu projekta opremanja laboratorija za napredno računanje na matematičkom odjelu Prirodoslovno Matematičkog Fakulteta u Zagrebu pod vodstvom profesora Nenada Antonića. Mnogi primjeri opisani u ovom radu dijelovi su sustava korištenog u tom projektu.

2. Operacijski sustav

2.1. Kratka povijest UNIX i Linux operacijskih sustava

Povijest UNIX operacijskog sustava ide unatrag sve do 1969. godine kada je UNIX začeo kao znanstveni projekt u AT&T Bell Labs. 1976. godine UNIX je postao dostupan sveučilištima bez naplate i time postao osnova velikog broja klasa operacijskih sustava te akademskih znanstvenih projekata.

Projekt Berkley UNIX počeo je 1977. godine kada je Computer Systems Research Group (CSRG) na Kalifornijskom sveučilištu Berkley kupio pravo korištenja izvornog koda od AT&T. Berkleyevi produkti (zvani BSD, po Berkley Software Distribution) počeli su 1977. godine s 1BSD za PDP-11, a posljednji je izdan 4.4.BDS 1993. godine.

Kako je UNIX dobivao na svojoj komercijalnoj prihvatljivosti, cijena pristupa izvornom kodu je naglo rasla. Konačno je Berkley krenuo za svojim dugoročnim ciljem da iz BSD-a izbací sav AT&T-ev izvorni kod, što je bio mukotrpan i dugotrajan proces. Prije nego što je taj projekt završen, Berkley je izgubio financije za svoje proučavanje operacijskih sustava te je CSRG raspušten.

Prije raspuštanja CSRG je pustio u javnost svoju konačnu zbirku svog ‘očišćenog’ izvornog koda, poznatog općimenom 4.4BSD-Lite. Većina današnjih verzija BSD UNIX operacijskih sustava (uključujući BSD/OS, FreeBSD, NetBSD i OpenBSD) potekla je iz 4.4BSD-Lite.

Preostalih verzija UNIX operacijskih sustava koje se danas mogu naći u upotrebi (npr. HP-UX, Solaris i IRIX) najvećim su dijelom potekle su od izvornog AT&T UNIX operacijskog sustava, dok Linux operacijski sustav nalazi svoje mjesto negdje između te dvije skupine.

Linux je nastao 1991. godine kao osobni projekt Linusa Torvaldsa, tada Finskog studenta na dodiplomskom studiju na Helsinškom sveučilištu. On je originalno zamislio svoj projekt kao skromnu varijantu Minix operacijskog sustava, razvijenog kao primjer operacijskog sustava od strane Andrew S. Tanenbauma, ali Linux je izazvao pozamašan interes u svijetu te je njegova jezgra ubrzo poprimila svoj vlastiti život. Iskorištavajući moć kooperacijskog razvoja Torvalds je nakon tog trenutka bio u mogućnosti nositi se s puno ambicioznijim planovima. Linux jezgra verzija 1.0 puštena je u javnost 1994. godine, a u trenutku ovog pisanja (kraj 2002. godine) posljednja stabilna verzija Linux jezgre je 2.4.19.

2.2. Veza UNIX i Linux operacijskih sustava

Linux je verzija UNIX operacijskog sustava. To je činjenica koju se zaista ne može previše naglasiti iako bi to neki nazvali svetogrđem. Linux je potpuno nova izvedba POSIX standarda koja radi na mnoštvu različitih platformi te je kompatibilna s većinom postojećih UNIX programa. Razlikuje se od većine drugih verzija UNIX-a po tome što je besplatan, cijeli mu je izvorni kod otvoren javnosti te je kooperacijski razvijen, pri čemu su njegovom razvoju doprinosile stotine pojedinaca i organizacija.

Linux zajednica često umanjuje važnost veze između Linux i UNIX operacijskih sustava kako bi istakla neovisnost razvoja Linuxa. Linux je zaista različit od UNIX operacijskog sustava po svojoj implementaciji i

načinu na koji se njome upravlja, tako da je razumljivo da ljudi koji se bave razvojem Linux operacijskog sustava te programa koji rade pod njime žele biti priznati kao nešto više od 'ljudi koji su još jednom klonirali UNIX'. Ipak, razlike između UNIX i Linux operacijskih sustava većinom se svode na najčešće nevažne detalje sa stajališta korisnika i administratora. Zapravo, Linux sustavi se općenito ponašaju više kao tradicionalni UNIX operacijski sustavi nego li što to rade neke današnje varijante UNIXa kao što su Sunov Solaris ili IBMov AIX.

Linux programi su UNIX programi. Zahvaljujući većinom GNU projektu, većina važnih programa koji daju UNIX operacijskim sustavima njihovu vrijednost razvijena je u nekom obliku modela 'otvorenog izvornog koda'. Taj isti izvorni kod radi kako na ne-Linux tako i na Linux operacijskim sustavima. Npr. Apache Web Server, često korišten serverski program za WWW sustav, radi u cjelosti bilo da je pokrenut na Linux ili na HP-UX operacijskom sustavu.

Također je bitno spomenuti da Linux nije jedina besplatna varijanta UNIX operacijskog sustava na svijetu. FreeBSD, NetBSD i OpenBSD, svi do jednog izvedeni iz originalne Berkley programske distribucije (Berkley Software Distribution), imaju velik broj svojih vjernih korisnika. Ti su operacijski sustavi uglavnom usporedivi s Linux operacijskim sustavom po svojim svojstvima i pouzdanosti, iako imaju nešto manju podršku od strane raznih tvrtki koje se bave proizvodnjom programa.

UNIX sustavi koriste se kao produkcijske okoline već dugi niz godina. Sada je Linux sazio do točke kada se i on također može koristiti kao operacijski sustav u takvoj okolini. Tematika kojom se bavi ovaj rad najprimjenjivija je baš u takvoj okolini, iako ima svoje primjene i na računalima s jednim korisnikom.

2.3. Linux distribucije

Linux projekt razlikuje se od ostalih varijanti UNIXa po tome što definira samo jezgru operacijskog sustava. Ta jezgra, zajedno s naredbama, daemonima te ostalom potrebnom programskom podrškom čini potpun i iskoristiv operacijski sustav, u Linux terminima – distribuciju. Sve Linux distribucije dijele iste jezgre, dok se sav dodatni materijal koji dolazi uz jezgru može uvelike razlikovati među različitim distribucijama.

Tablica 2-1: Većina popularnih Linux distribucija opće namjene

Distribucija	WWW stranice	Komentar
BestLinux	www.bestlinux.net	Podržava više jezika, laka instalacija
Caldera	www.caldera.com	Orijentirana poslovnim svijetom i stabilnosti
Debian	www.debian.org	Najbliža GNU projektu
Linux Mandrake	www.mandrake.com	Optimizirana ali bez gubitka sposobnosti
Red Hat Linux	www.redhat.com	Najčešće korištena distribucija opće namjene
Slackware Linux	www.slackware.com	Nekad dominantna distribucija
SuSE Linux	www.suse.com	Posebno jaka u Evropi, podržava više jezika
TurboLinux	www.turbolinux.com	Jaka u Aziji, podržava Azijske jezike

Distribucije su navedene abecednim redom, a ne po važnosti.

Mnoge manje distribucije nisu navedene u gornjoj tablici. Osim njih postoji mnoštvo navedenih distribucija namijenjenih grupama sa specijaliziranim potrebama, kao što su ljudi koji se bave razvojem programa na ugrađenim

sustavima. Puno opširniji vodič kroz Linux distribucije, uključujući mnoge kojima osnovni korisnički jezik nije engleski, može se naći na WWW stranicama <http://www.linux.org/dist>.

2.4. Debian GNU/Linux distribucija

Već najpovršnijim pregledom po internetu uočava se kao daleko najčešće postavljano pitanje vezano uz Linux te najrjeđe odgovoreno upravo 'Koja je najbolja Linux distribucija?'. Nažalost, niti u ovom tekstu ne može se naći odgovor na to pitanje. Pravi odgovor ovisi o tome kako je zamišljeno da se sustav koristi, o varijantama UNIX operacijskog sustava s kojima smo upoznati, važnosti tehničke podrške za odabrani operacijski sustav pa čak i o političkom usmjerenju.

Većina Linux distribucija može napraviti sve što bi smo ikada mogli zahtijevati od Linux sustava. Neke od njih zahtijevaju instalaciju dodatne programske podrške da bi bili potpuno funkcionalni, dok drugi mogu zahtijevati neke druge postupke. Bilo kako bilo, razlike među njima nisu od nepremostive važnosti. U biti, misterij je zašto postoji toliki broj različitih distribucija, od kojih svaka tvrdi da je 'laka instalacija' i 'velika programska podrška' upravo njihova osnovna karakteristika. Teško je ne doći do zaključka da ljudi jednostavno vole stvarati nove Linux distribucije.

Prihvatanjem jedne Linux distribucije zapravo ulažete u jedan specifičan način kako određena strana – održavatelj te distribucije – organizira operacijski sustav. Umjesto gledanja samo na sposobnosti instalirane programske podrške, mudrije je pažljivo razmotriti kako će vaša organizacija i održavatelj odabrane distribucije raditi zajedno u godinama koje slijede. Neka od bitnih pitanja koja bi bilo dobro postaviti su:

- Hoće li ova distribucija uopće postojati za pet godina?
- Hoće li ova distribucija biti u toku s posljednjim sigurnosnim zakrpama na sustavu?
- Hoće li ova distribucija redovno izdavati nove verzije programske podrške?
- Ako imam problem, hoće li održavatelj razgovarati sa mnom?

Kad stvari pogledamo pod ovim svjetlom, onda mnoge manje Linux distribucije, izvan glavnih smjerova razvoja, prestaju biti zanimljive. S druge strane ne moraju sve uspješne Linux distribucije imati iza sebe veliku organizaciju.

Debian GNU/Linux je primjer Linux distribucije s potpuno drugačijim pristupom održavanju i razvoju te Linux distribucije od ostalih, njemu paralelnih Linux distribucija. Iza nje ne stoji jedna kompanija već je ta distribucija puno otvorenija i na njoj radi velik broj ljudi diljem cijelog svijeta, čime je ta Linux distribucija automatski izrazito aktivna te za nju postoji i lako je na velika količina tehničke i programske podrške. Kao dodatne dobre strane Debian GNU/Linux distribucije možemo navesti vrlo razvijen sustav za upravljanje programskim paketima koji omogućava laku instalaciju i održavanje, kako pojedinačnih tako i složenih sustava sastavljenih od mnoštva radnih stanica. Prije izdavanja nove verzije vodi se strogo testiranje svih dijelova sustava čime se osigurava najveća moguća pouzdanost te je sav razvoj sustava povezan kroz javno dostupan sustav za praćenje pogrešaka i napretka što pruža vrlo jednostavan način kojim

korisnici Debian GNU/Linux distribucije lako i brzo mogu doći do povratnih informacija.

3. Standardna instalacija operacijskog sustava

3.1. Tradicionalna instalacija

Kod instalacije operacijskog sustava postoji velik broj koraka koji mora biti obavljen. Sam niz koraka razlikuje se od računala do računala ovisno o njegovoj arhitekturi te namjeni.

Tradicionalno instalacija jednog operacijskog sustava odvijala se tako da se moralo sjesti za računalo, jedan po jedan obaviti sve korake te na računalo instalirati sve potrebne dijelove operacijskog sustava i potrebnu programsku podršku. Cijeli postupak odvijao se od početka za svako računalo te zahtijevao veliku spremnost (pa i cijenu) administratora sustava kako bi mogao kod svakog računala taj postupak prilagoditi njegovim specifičnostima.

Kao posljedica postupak je bio dugotrajan (instalacija pojedinog računala može trajati satima, a trajanje instalacije jednog računalnog sustava raste barem proporcionalno broju članova tog sustava), odgovornost administratora velika, a veći sustavi računala nestabilni uslijed raznih neskladnosti među pojedinim dijelovima u sustavu.

Velik problem za organizaciju osnovanu na takvom računalnom sustavu je ogromna cijena tog sustava, kao i velika tromost te neprilagodljivost cijelog sustava potrebama organizacije uslijed stalnog rada na njegovom održavanju. Zbog takve tromosti i eksponencijalnog rasta troškova računalni sustavi postavljeni ovom metodom ograničeni su u veličini do koje mogu rasti.

Uz sve to, postoji i problem nakupljanja programskih sredstava potrebnih za izvođenje same instalacije. Prilikom svake instalacije računala administrator bi trebao proanalizirati same komponente od kojih je fizički sastavljeno računalo te prije instalacije pripremiti medije (CD, disketa) s odgovarajućom programskom podrškom. S obzirom na to da se zahtjevi različitih komponente uvelike razlikuju, količina materijala potrebna za održavanje jedne malo veće količine računala lako je mogla zauzeti cijelo skladište, a administrator bi morao imati izrazito velike organizacijske sposobnosti kako bi bio sposoban nositi se s tolikom količinom fizičkog posla i materijala.

3.2. Mrežna instalacija

Za unaprjeđivanje tradicionalnog postupka instalacije mreže računala bilo je nužno obaviti par bitnih koraka:

- Prikupiti potrebne programe na jedno centralizirano spremište
- Omogućiti lak pristup spremištu potrebnih programa s računala koje želimo instalirati

Uz korištenje današnjih mreža računala te je preduvjete lako zadovoljiti tako da se sva potrebna programska podrška centralizira na jednom računalu, a kroz mrežu dopusti lak pristup do potrebnog materijala.

Ovime je već postignuto mnogo. Administratori sustava su mogli prestati imati ulogu skladištara, prostor koje je to tada zauzimao potreban materijal u raznim ormarima i skladištima smanjen je na prostor potreban za čuvanje jednog ili nekolicine računala, a uvelike je olakšan dostup te pregled raspoloživog materijala.

Sada kada administrator nije više morao sa sobom vući potreban materijal sljedeći logičan korak bilo je omogućavanje instalacije s jednog centralnog mjesta. Za to je bilo potrebno također ostvariti par preduvjeta:

- Omogućiti pristup računalu kroz mrežu prije nego što je na računalo instaliran bilo kakav programski sadržaj
- Omogućiti programsku analizu fizičkih komponenti računala

Prvi dio rješen je prirodnim razvojem komponenti za umreživanje računala. Naime, osnovne fizičke komponente počele su se proizvoditi s ugrađenom programskom podrškom za uključivanje računala preko mreže te za omogućavanje inicijalnog pristupa na još neinstalirano računalo. Točan mehanizam kojim se to omogućuje bit će detaljnije opisan kasnije u togu ovog rada. Osim toga standardizirane su neke klase komponenti čime je omogućeno programsko analiziranje pojedinih komponenti računala te automatsko određivanje programske podrške potrebne za instalaciju konkretnog računala.

Sve to zajedno u ogromnoj mjeri olakšava posao administratora koji je sada u mogućnosti, nakon što je novo računalo fizički spojeno na mrežu, pristupiti tom računalu bez udaljavanja sa svog radnog mjesta, s tog mjesta ga proanalizirati te na njega instalirati svu potrebnu programsku podršku.

Ovom reorganizacijom načina instalacije posao je postao uvelike lakši i jeftiniji, ali i dalje nedovoljno definiran da bi doživio potpunu automatizaciju.

3.3. Debian GNU/Linux instalacija

Lijep primjer mrežne instalacije može se vidjeti u standardnom mehanizmu korištenom za instalaciju Debian GNU/Linux distribucije operacijskog sustava na računala u mreži.

Cijeli operacijski sustav te cijela njegova programska podrška sadržana u ovoj distribuciji jasno je i detaljno razdijeljena u odvojene pakete. Svi ti paketi lako su dostupni s jednog centraliziranog mjesta na internetu. Osim toga jasno je definiran postupak kojim se to centralno mjesto za Debian GNU/Linux distribuciju može duplicirati te održavati na dodatnim mjestima. Time je riješen problem eventualne preopterećenosti pristupa centralnom spremištu.

Za pristup do tog mjesta osigurani su standardni programi i to na nekoliko razina. Svaka razina koristi prethodne te dodaje veću prilagođenost sustava krajnjem korisniku. Najniža razina čini program apt-get koji omogućava preciznost dovoljnu za skriptiranje i automatiziranje dijelova procesa instalacije, dok se na višoj razini nalaze razni korisnički programi (npr. dselect) koji korisniku omogućavaju baratanje Debian paketima kroz intuitivno grafičko sučelje.

Debianov sustav paketa koristi sustav 'povezanosti' među paketima koje su pažljivo dizajnirane od strane održavatelja Debian paketa. Te povezanosti omogućuju iskazivanje veza među paketima kao što su npr:

- *Ovisi (depends)*: paket A ovisi o paketu B znači da paket B apsolutno mora biti instaliran na računalo kako bi sadržaj paketa A mogao ispravno funkcionirati. Preciznije: paket A ovisi o točnoj verziji paketa B, tj. ovom povezanošću forsirana je i verzija paketa B koja mora biti instalirana na računalo.

- *Savjetuje (recommends)*: paket A *savjetuje* instaliranje paketa B ako održavatelj paketa A smatra da većina korisnika ne bi željela sadržaj paketa A bez da ima na računalu instaliran također i sadržaj paketa B.
- *Predlaže (suggests)*: paket A *predlaže* paket B ako je sadržaj paketa B povezan sa sadržajem paketa A te ga najčešće unaprjeđuje.
- *Sukobljava (conflicts)*: paket A se *sukobljava* s paketom B ako sadržaj paketa A neće funkcionirati na sustavu ukoliko je na njemu instaliran i paket B.
- *Zamjenjuje (replaces)*: paket A *zamjenjuje* paket B ukoliko instalacija paketa A uklanja te u nekim slučajevima zamjenjuje datoteke instalirane paketom B.
- *Sadrži (provides)*: paket A *sadrži* paket B ako su sve datoteke i funkcionalnost paketa B sadržani paketu A. Ovaj mehanizam omogućuje korisnicima s ograničenim prostorom na disku da dohvate samo dijelove paketa A koji im zaista trebaju.

Te povezanosti su dokumentirane sa svakim paketom i to na standardan način čime je omogućeno njihovo automatsko provođenje i kontroliranje.

Primjer:

Debian paket *gcc* koji sadrži GNU C prevodilac ovisi o paketu *binutils* koji sadrži neke dijelove potrebne za rad prevodioca. Ukoliko korisnik želi u sustav instalirati paket *gcc* bez da je prije toga na njega instalirao paket *binutils*, sustav za baratanje paketima će na to upozoriti korisnika te mu omogućiti automatsko instaliranje odgovarajućih potrebnih paketa.

Kao pomoć pri manipulaciji Debian paketima razvijeni su alati koji olakšavaju aktivnosti kao što su:

- Baratanje paketima ili nekim njihovim dijelovima.
- Razbijanje jednog ili više Debian paketa na dijelove kako bi se mogli prenjeti medijem ograničene veličine, npr. disketama.
- Konstruiranje novih Debian paketa.
- Instalacija paketa s nekog odvojenog mjesta (drugog računala) na mreži korištenjem HTTP, FTP ili NFS protokola.

Osim samog sadržaja koji treba biti instaliran na neko računalo Debian paketi sadrže još jednu izrazito korisnu stvar, a to su skripte kojima se može upravljati postupkom instalacije samog paketa. Tako je unutar samog paketa moguće ugraditi podešavanja osnovnih konfiguracijskih datoteka cijelog sustava kako bi sadržaj paketa mogao ispravno raditi.

Cijeli ovaj mehanizam koji je Debian GNU/Linux distribucija postavila za instaliranje operacijskog sustava na klijent mi ćemo iskoristiti pomoću kasnije opisanog FAI sustava da bismo ostvarili posve automatsku instalaciju računala na mreži.

3.4. Automatizirana mrežna instalacija

Sva prethodna unapređenja postupka instalacije operacijskog sustava dala su nam čvrstu osnovu, ali nam i dalje nisu uklonili potrebu da čovjek (administrator) prilikom svake instalacije dio posla obavlja ručno. Istina, opseg tog posla uvelike je smanjen od inicijalnih tradionalnih instalacija

operacijskih sustava, ali čitav mehanizam ima dijelove koji i dalje zahtijevaju čovjekovu osobnu aktivnost. Ta aktivnost još ni danas ne može biti potpuno uklonjena, ali ima dijelova gdje može biti i dalje smanjena.

Npr. ako na fakultetu postoji učionica s dvadeset istovrsnih računala na kojima studenti odrađuju dio svog nastavnog programa, onda administrator te mreže računala mora relativno često reinstalirati ta računala, eventualno s posve različitim operacijskim sustavima – kako bi se predviđeni nastavni plan i program mogli nesmetano odvijati. Dosadašnjim unapređivanjem postupka instalacije operacijskih sustava uspjeli smo postići da administrator taj posao može obaviti sjedeći samo za jednim računalom u toj mreži, ali to i dalje nije dovoljno. Naš cilj je da administrator samo jedanput definira tok instalacije jednog operacijskog sustava, da sva kasnija reinstaliranja tog istog operacijskog sustava teku potpuno automatizirano, a da, ako je moguće, i ono inicijalno instaliranje može u velikoj mjeri iskoristiti posao već obavljen na drugim instalacijama. Time bi danas zahtjevan posao administriranja jedne studentske računalne mreže bio u potpunosti automatiziran i sama administracija tih računala ne bi predstavljala kočnicu za daljnji razvoj sveučilišta kao ustanove – povećanju broja učionica, sudenata, računala i slično.

U posljednjem stupnju evolucije postupka instalacije operacijskih sustava dolazi se do potpune automatizacije. Posljedice ovakve organizacije ogromne su i od neprocjenjive vrijednosti. Time je trošak administriranja operacijskim sustavima na jednoj mreži računala postao konstantan, a nije proporcionalan broju računala u toj mreži. Osim toga, smanjenjem ljudske uloge u samom postupku uvelike je smanjena količina nepravilnosti i pogrešaka u sustavu, a baratanje sustavom je mnogostruko ubrzano.

Jedna zgodna nit vodilja za ovaj razvoj instalacije operacijskih sustava je ova: 'Ne želimo ručno postavljati male postavke na svakom pojedinom računalu naše mreže. To je dosadno i definitivno postoje puno pametnije i interesantnije stvri koje ljudi mogu raditi, a uz to nitko nikada ne može popamtiti sve male ručne promjene koje su bile napravljene na velikom i važnom računalu u mreži nakon što mu se pokvari disk te mu se ugradi novi koji opet treba instalirati da radi isto kao stari – i to hitno.'

4. Opis općeg sustava za automatizirano centralno održavanje mreže računala

4.1. Osnovni dijelovi sustava

Osnovni dijelovi našeg sustava za automatizirano centralno održavanje mreže računala su:

- Gold server – centralno mjesto za upravljanje našom mrežom računala
- Install serveri – 'skladišta' velikih programskih paketa potrebnih za instalaciju
- Klijenti – računala u mreži na kojima se vrši instalacija.

Svi ti dijelovi bit će detaljnije opisani u narednim odjeljcima.

4.1.1. Gold server

Gold server je računalo koje predstavlja mozak naše mreže računala. To je jedino mjesto na kojemu se konfiguriraju instalacije pojedinih računala te na kojem je potrebno raditi ručne promjene. Da bi cijeli sustav glatko funkcionirao ovo pravilo moramo prihvatiti i sljepo ga se pridržavati. Svaka ručna promjena na bilo kojem drugom dijelu sustava uvodi nepravilnosti u sustav te onemogućava automatizaciju postupka i u globalu vraća nas korak nazad u razvoju. Ukoliko se ikada prilikom održavanja ovakvog sustava nađemo pred problemom čije nas predloženo rješenje vodi k tome da ručno podešavamo postavke na bilo kojem drugom dijelu cijelog sustava onda je vrijeme da ponovo razmotrimo to rješenje te njegove štetne posljedice.

Ovo nam omogućava trenutno reproduciranje bilo kojeg dijela naše mreže kad god se za to ukaže potreba. Npr. ukoliko sveučilište odluči preurediti studentski računalni praktikum tako da se na računalima u njemu može održavati konferencija, onda nakon te konferencije cijeli praktikum može biti vraćen u originalno stanje u roku ne duljem od 20 minuta bez obzira na broj računala kojima je bila privremeno promijenjena namjena.

Odavdje se upravlja svim dijelovima mreže neovisno, tj. nije potrebno imati poseban server za upravljanje različitim dijelovima mreže ovisno o njihovoj namjeni. Ovakav pristup uvelike olakšava administriranje cijelim sustavom.

4.1.2. Install serveri

Kod manje mreže Gold server može, osim što upravlja mrežom, također opskrbljivati sve klijente potrebnim programskim paketima, dok kod većih mreža takav previše centraliziran pristup može pretvoriti pristup Gold serveru u usko grlo cijelog sustava. Ovakav bi nas problem ozbiljno zaskočio u slučaju kad bi u isto vrijeme u nekoj zgradi bilo pušteno u pogon stotinjak novih neinstaliranih klijenata koji bi svi odjednom 'napali' Gold server sa zahtjevima za instalacijom. Sami zahtjevi ne bi predstavljali veći problem uz današnje brzine računala i moderne mrežne opreme kojom su povezani, ali svaka instalacija zahtjeva da se klijentu dostavi ogromna količina programskih paketa što bi u ovakvoj situaciji potpuno zatvorilo pristup Gold serveru.

Kako bi se riješio ovaj problem Gold servera se oslobađa uloge spremišta koja se onda dodjeljuje nizu drugih računala čija je zadaća isključivo čuvanje programskih paketa te njihovo dostavljanje na zahtjev klijenata. Gold server i dalje zadržava ulogu upravljačkog centra, ali umjesto da klijentima pruži odmah potrebne programske pakete on im proslijedi informaciju o tome koji im programski paketi trebaju te na kojem Install serveru ih mogu tražiti.

Ovakvo rješenje omogućava ravnomjernu raspoređenost opterećenosti cijelog sustava te omogućava cijelom sustavu mogućnost neograničenog rasta – što od samog početka i je naš cilj.

4.1.3. Klijenti

Klijentima nazivamo sva ostala računala u našem sustavu. Kao što je gore jasno naglašeno, izrazito je bitno da se svo održavanje svakog klijenta obavlja kroz Gold server.

Postoji još jedan pristup koji je potrebno usvojiti da bi sustav savršeno funkcionirao. To je takozvani ‘vuci, a ne guraj’ pristup, tj. pravilo da nikada u toku rada sustava servera ne smije klijentu nametati neku promjenu ili instalaciju programskog paketa, već klijent mora uvijek biti taj koji traži informacije o instalaciji te dijelove potrebne za njeno provođenje. Ovakav pristup je također nužan kako bi veliki sustavi mogli funkcionirati, a već i na manjim sustavima dosljedno držanje ovog pravila automatski održava red unutar sustava. Problem s obrnutim pristupom bio bi taj da bi Gold server u svakom trenutku morao točno znati što se nalazi na pojedinim klijentima, a to nije moguće jer do promjena na klijentu može doći lokalno – bez znanja Gold servera. Kao posljedica cijeli bi sustav bio izrazito krhak i podložnam pogreškama.

Kao podrška administratorima (neka vrsta posljednje linije obrane) uvijek postoji mogućnost nametanja neke promjene klijentu (osim naravno urednog centralnog gašenja i paljenja svih računala u velikim mrežama), ali takve mjere moraju uvijek biti korištene s oprezom i nikako ne smiju postati dio redovnih aktivnosti.

Kao primjer gdje bi nadržanje ovog pravila ili njegovo nedosljedno provođenje dovelo do problema te nepopravljivog degeneriranja sustava opet možemo dati studentski računalni praktikum na fakultetu. To je okolina gdje su sva računala konstantno pod kontrolom, često i ne baš dobronamjerne, grupe studenata u kojoj se uvijek nađe nekolicina koja neumorno pronalazi razloge zašto bi pokušali cijeli sustav barem djelomično onesposobiti, ako ne i potpuno zaustaviti. Ukoliko bi tada neki klijent obrisao dio operacijskog sustava, npr. program potreban za održavanje nastave za sljedeću skupinu studenata onda Gold server koji ‘gura’ promjene na klijenta ne bi mogao znati da se to desilo te ne bi mogao tu pomjenu ispraviti. Ukoliko bi iz nekog razloga čak i odlučio ponovo instalirati odgovarajući programski paket na klijente (npr. ukoliko je postavljeno da se svi bitni programski paketi potrebni za nastavu ponovo instaliraju neposredno prije održavanja vježbi na kojima su potrebni) nitko ne bi mogao garantirati da baš tada student nije ugasio promijenjenu mašinu pa da na njoj ne mogu biti napravljene potrebne promjene.

U slučaju pridržavanja gornjeg pravila, da klijenti trebaju pitati za sve promjene na njima, do ovakvih komplikacija ne bi moglo doći. Točno kako taj

pristup izbjegava gore navedene probleme može se vidjeti u sljedećem opisu rada cijelog sustava.

4.2. Održavanje kontrole nad sustavom

Kako bi Gold server mogao održavati kontrolu nad cijelim sustavom svaki klijent svoj originalni operacijski sustav podiže s mreže. Klijent prilikom svog paljenja na mrežu šalje signal kojim javlja Gold serveru da se on pali i traži svoj inicijalni operacijski sustav. Ukoliko Gold server iz bilo kojeg razloga nije dostupan, na klijentu se diže njegov lokalni operacijski sustav, ukoliko takav uopće na njemu postoji. Nakon što Gold server primi signal o buđenju klijenta, šalje mu potrebne informacije. Te informacije uključuju:

- Informacije o samom klijentu: njegovu adresu na mreži, adrese različitih servera na mreži dostupnih tom klijentu, adrese potrebne za pristup mrežnim printerima i sl.
- Informacije o inicijalnom operacijskom sustavu koji treba biti podignut preko mreže na klijentu: točna identifikacija tog operacijskog sustava, dodatnih programskih paketa potrebnih za njegov rad te popis Install servera od kojih je moguće dobiti potrebne podatke.

Nakon što je primio potrebne informacije klijent od Install servera traži svoj inicijalni operacijski sustav koji mu se dostavlja preko mreže te se na klijentu i pokreće. Primjer mehanizma koji omogućuje ovakvo dizanje operacijskog sustava na klijentu bez korištenja klijentovog lokalnog diska dan je u dodatku na kraju ovog rada. Cilj tog operacijskog sustava je da na brzinu provjeri klijentov lokalni sustav te od Gold servera zatraži sve potrebne nadogradnje i ispravke. Npr. ukoliko na tom klijentu još uvijek ne postoji lokalno instalirani operacijski sustav pokreće se njegova instalacija. Tu se također odvijaju provjere te eventualni ispravci konfiguracijskih datoteka na klijentu potrebnih za njegov normalan rad. Ovaj postupak čini sustav otporan na pogreške izazvane lokalnim (čak i zlonamjernim) izmjenama na klijentu.

Nakon što je taj inicijalni operacijski sustav odradio svoja zaduženja klijent podiže svoj lokalni operacijski sustav i potpuno je spreman za rad.

5. FAI sustav

5.1. O FAI sustavu

Kao što je već ranije u ovom radu opisano, standardni mehanizam za instalaciju Debian GNU/Linux sustava uvelike olakšava rad administratora sustava zaduženog za instalaciju, ali i dalje sam po sebi ne omogućava automatsko provođenje cijelog postupka, a automatizacija tog postupka sama se prirodno nametnula kao iduća evolucijska stepenica.

FAI (Fully Automatic Installation) je neinteraktivni sustav za instalaciju Debian GNU/Linux operacijskog sustava na jedno računalo ili skupinu računala bez potrebe za ljudskim intervencijama. To je sustav razvijen te se i dalje razvija i nadograđuje na Institutu za informatiku Sveučilišta u Kölnu u Njemačkoj pod vodstvom Thomasa Langea, sistem administratora na tom institutu, a nama će ovdje poslužiti za detaljno razmatranje jednog sustava za centraliziranu automatsku instalaciju operacijskih sustava.

Rad na FAI sustavu započeo je 2001. godine, a prva verzija FAI-a izdana je već iste godine. Danas je FAI uključen kroz nekoliko Debian paketa u sklopu standardne Debian GNU/Linux distribucije. S tom distribucijom trenutno dolazi verzija FAI sustava 2.3.1., a posljednja aktivna verzija FAI sustava koja je trenutno stabilna i puštena u javnost je 2.3.4.

5.2. Metoda razvoja FAI sustava – open source

Zanimljivost te možda snaga FAI sustava izlazi i iz činjenice da je cijeli sustav razvijen u duhu Linuxa metodom ‘otvorenog izvornog koda’ (open source), tj. tokom njegovog razvoja cijeli izvorni kod projekta bio je dostupan svakome tko je bio zainteresira. Tako je cijeli projekt vrlo brzo dobio mnoštvo poklonika koji su ulagali svoje vrijeme i time aktivno radili na razvoju tog sustava. Među svim sudionicima u razvoju FAI sustava uspostavljena je komunikacija putem mailing listi na internetu, a razvoj cijelog sustava može se pratiti u centralnom CVS (Concurrent Versions System) spremištu preko interneta. Svi prijedlozi, nadogradnje i ispravke vezani uz FAI kod šalju se originalnom autoru na Institut za informatiku Sveučilišta u Kölnu, koji ih sa svojim timom pregledava te ubacuje kao službeni dio FAI sustava.

5.3. Posebnosti korištene implementacije FAI sustava

FAI sustav se skoro savršeno uklapa u naš model razvijen prilikom analize općeg sustava za centraliziranu automatsku instalaciju operacijskih sustava, a ovdje ćemo navesti neke njegove važnije posebnosti.

- Prva je čisto kozmetičke prirode, a ta je da se Gold server u sklopu FAI sustava naziva i FAI server.
- Druga je da smo se za potrebe ovog rada ograničili na to da nam Gold server bude i jedini Install server u sustavu iako FAI ne postavlja to kao strogo ograničenje.
- Treća i vrlo bitna posebnost FAI sustava je ta da trenutno u službenoj verziji FAI sustava ne postoji unaprijed ugrađena metoda kojom se na

klijentima može unaprjeđivati njegov operacijski sustav iako je ostavljen jasan prostor gdje se takvo proširenje može nadograditi.

- Posljednja posebnost je zapravo jedno ograničenje samog Linux operacijskog sustava. A to je da trenutno Linux operacijskom sustavu nije moguće ‘uredno’ programski promijeniti mjesto odakle će se idući puta podići operacijski sustav na klijentu. To znači da ukoliko klijent podigne svoj operacijski sustav preko mreže onda nije moguće ‘uredno’ automatski podignuti klijentov lokalno instalirani operacijski sustav. Postoji ‘zakrpa’ za ovaj problem, ali ona, kao i svaka zakrpa u računarstvu, nosi sa sobom drugi skup problema. Detaljniji opis ovog slučaja bit će obrađen malo kasnije.

Ono što predstavlja veliku prednost FAI sustava je ta da svi ovo nedostaci i nadogradnje, s obzirom na velik broj ljudi koji aktivno rade na poboljšanju FAI sustava, već postoje rješeni i javno dostupni putem interneta. Još jedino čekaju da budu provjereni i ubačeni u službenu verziju FAI sustava od strane Instituta za informatiku Sveučilišta u Kölnu.

5.4. Prvi pogled na rad FAI sustava

5.4.1. Osnovne postavke FAI servera

Da bi FAI server mogao uopće započeti obavljati svoju ulogu na njemu moraju biti prije toga instalirani i pokrenuti programi koji omogućuju korištenje nekih nužnih mrežnih protokola.

Kako bi klijent inicijalno pri svom buđenju kontaktirao FAI server koristi neki od mrežnih protokola za konfiguriranje računala preko mreže. Primjeri takvih protokola su BOOTP (BOOT Protocol) i DHCP (Dynamic Host Configuration Protocol). Ovisno o tome za koji smo se odlučili na FAI serveru moramo instalirati i pokrenuti odgovarajući server program (daemon). U našem slučaju koristili smo DHCP protokol. Detaljniji opis BOOTP i DHCP protokola može se naći u dodatku ovog rada.

Drugo, na FAI serveru mora klijentima biti omogućen pristup do nekih direktorija potrebnih za rad FAI sustava. To se postiže instaliranjem NFS server programa. NFS (Network File System) je mrežni protokol koji omogućava računalu u mreži pristup podacima spremljenim na drugo računalo kao da se oni nalaze na lokalnom disku. Takav postupak naziva se *mountanje* diskova preko mreže – *disk mounting*.

I treće, ukoliko se, kao u našoj implementaciji FAI sustava, FAI server koristi i kao Install server onda je na njemu potrebno pokrenuti program koji omogućava korištenje TFTP (Tiny File Transfer Protocol) protokola. Taj se protokol koristi za prijenos jezgre FAI mrežnog operacijskog sustava s FAI servera na klijenta.

Sada je još prije same instalacije klijentovog operacijskog sustava na serveru potrebno definirati točno što i kako mora biti instalirano na klijentu od programske podrške. FAI sustav omogućava, kao što će u daljnjem dijelu ovog rada biti pobliže opisano, vrlo detaljno specificiranje svakog aspekta te instalacije koristeći sustav klasa kojima može pripadati pojedini klijent.

5.4.2. Podizanje FAI klijenta preko mreže

Klijenti na kojima treba, koristeći FAI sustav, instalirati operacijski sustav ili samo neki njegovi dijelovi prvo trebaju preko mreže kontaktirati FAI server.

Ukoliko se u klijentu model mrežne kartice koji na sebi nema ugrađenu podršku za mrežno podizanje operacijskog sustava onda je moguće inicijalno podizanje sustava obaviti tako da se sustav podigne s posebno pripremljene diskete na kojoj se nalazi program koji će onda iskoristiti mrežnu karticu u klijentu da se dalje obavi normalno podizanje operacijskog sustava preko mreže. Takav program za podizanje računala s diskete moguće je nabaviti besplatno na internetu na adresi <http://www.rom-o-matic.net>. Točan program koji se koristi ovisi o tipu mrežne kartice koja se nalazi u klijentu, ali na ovim WWW stranicama dostupni su programi za većinu danas postojećih modela mrežnih kartica. Zgodno je napomenuti da su ti programi također razvijeni modelom 'otvorenog izvornog koda' te su kao takvi potpuno dostupni za proučavanje i daljnju nadogradnju.

Sada klijent putem odabarnog mrežnog protokola (BOOTP ili DHCP) dohvaća s FAI servera svoju IP adresu, jezgru FAI mrežnog operacijskog sustava korištenog tokom instalacije koristeći TFTP te direktorij na mrežnom datotečnom sustavu (NFS) koji će FAI mrežni operacijski sustav iskoristiti kao svoj osnovni (root) direktorij. Ovaj se mrežni direktorij u daljnjem tekstu zove *\$NFSROOT* direktorij.

Nakon što je na klijentu pokrenut FAI mrežni operacijski sustav kontrolu nad klijentom preuzimaju FAI skripte koje vrše daljnju automatsku instalaciju.

Na početku se na klijentu particioniraju i formatiziraju lokalni hard diskovi. Na njima se priređuju klijentovi lokalni datotečni sustavi te se instaliraju osnovni programski paketi standardnim Debian GNU/Linux instalacijskim sustavom. Nakon što su svi potrebni programski paketi instalirani na klijenta na njega dolazi i jezgra klijentovog novog operacijskog sustava. U biti ona je zapakirana u jedan Debian paket i instalira se potpuno identično kao i ostala programska podrška. Predposljednji korak je konfiguriranje klijentovog novog operacijskog sustava koje izvršavaju skripte pripremljene na FAI serveru te se na kraju klijent ponovo podiže sa svojim novoinstaliranim operacijskim sustavom – sve bez i jedne ljudske intervencije.

Izrazito je bitno naglasiti da kad je ovakva instalacija klijenta jedanput obavljena, njeno ponavljanje te time postavljanje klijenta nazad u početno stanje ne predstavlja nikakav problem i pruža izrazito elegantan sigurnosni mehanizam za proučavanje operacijskih sustava. Cijena eksperimentiranja s operacijskim sustavom te njegovog dovođenja u nedefinirano stanje iz kojega ga ne znamo vratiti je postala praktički nula. Deset minuta i računalo je opet u početnom stanju.

5.4.3. Spašavanje klijenata

Osim kao sustav za automatsku instalaciju klijenta FAI se može koristiti i za spašavanje podataka s klijentovih lokalnih diskova nakon što je iz bilo kojeg razloga klijentov lokalni operacijski sustav postao neispravan. Moguće je na klijentu samo podignuti FAI mrežni operacijski sustav bez pokretanja instalacije. Time se administratoru omogućuje pun pristup klijentovim

lokalnim hard diskovima bez podizanja klijentovog lokalnog operacijskog sustava.

5.4.4. Neke mogućnosti FAI sustava

- Potpuno automatska instalacija klijenata
- Podrška za BOOTP, DHCP i PXE boot protokole
- Nema potrebe za inicijalnim ramdiskom – 8MB memorije na klijentu je dovoljno
- Radi i na računalima s 386 procesorima
- Jezgra instalacijskog operacijskog sustava može koristiti dodatne module
- Moguće povezivanje s klijentom preko mreže u toku samo instalacije klijenta
- Dostup do dva dodatna konzolna terminala tokom instalacije
- Iste konfiguracije mogu biti jednostavno i prirodno korištene od strane klijenata sa sličnom namjenom
- Log datoteke za sva pokretanja FAI sustava na klijentu uredno se automatski spremaju na FAI serveru
- Podržava shell, perl, expect i cfengine skripte za dodatnu konfiguraciju klijenta
- Može pristupiti spremištu Debian distribucije preko NFS, FTP ili HTTP protokola
- Omogućava detaljno podešavanje FAI mrežnog operacijskog sustava korištenog tijekom instalacije te za spašavanje klijenata (kao npr. podešavanje rasporeda tipki na tipkovnici)
- Može se koristiti kao sustav za spašavanje podatka s klijenata
- Testiran na računalima SUN SPARC arhitekture s Linux i Solaris operacijskim sustavima
- Omogućava nadogradnju i izmjenu samog mehanizma instalacije tzv. 'hook' sustavom
- Podržava više različitih datotečnih sustava

5.5. Osnovni dijelovi FAI sustava

5.5.1. FAI server

Na FAI serveru postoji nekoliko dijelova bitnih za rad FAI sustava. To su redom:

5.5.1.1. Osnovna konfiguracijska skripta fai.conf

Nalazi se u direktoriju */etc/fai* i sadrži osnovne postavke potrebne za rad FAI sustava. Primjeri ovakvih postavki su lozinka za pristup root korisničkom računu na FAI mrežnom operacijskom sustavu te popis dodatnih Debian paketa koje želimo imati instalirane na tom operacijskom sustavu, a potpun popis ovih postavki bit će dan malo kasnije u dijelu rada koji govori o instalaciji FAI servera.

5.5.1.2. Osnovne skripte zadužene za održavanje same instalacije

Ove skripte nalaze se u direktoriju */usr/sbin*, a služe isključivo za održavanje same instalacije FAI sustava na FAI serveru te nemaju nikakve veze s konfiguriranjem instalacije pojedinih klijenata. FAI skripte koje se nalaze u ovom direktoriju u standardnoj verziji FAI sustava 2.3.4 su:

- *fai-setup*
- *make-fai-nfsroot*
- *make-fai-bootfloppy*

5.5.1.3. Kostur FAI sustava

Treći dio FAI sustava su skripte koje čine kostur potreban za rad FAI sustava. U njima je napravljena osnovna struktura koju prati svaka FAI instalacija. Na FAI serveru nalaze se u sljedećim direktorijima:

- */usr/lib/fai/sbin*
- */usr/share/fai*

5.5.1.4. \$NFSROOT direktorij

Četvrti osnovni dio FAI sustava na FAI serveru čini FAI-ev *\$NFSROOT* direktorij. Taj direktorij služi kao osnovna (root) particija za klijente dok je na njima podignut FAI mrežni operacijski sustav. Točnu putanju ovog direktorija određuje se postavljanjem vrijednosti odgovarajuće varijable u osnovnoj FAI konfiguracijskoj datoteci *fai.conf*, a po inicijalnim postavkama postavljenim u FAI sustavu putanja do tog direktorija je:

- */usr/lib/fai/nfsroot*

Sadržaj *\$NFSROOT* direktorija nije dozvoljeno ni u kojem slučaju ručno mijenjati već se njegov sadržaj konfigurira postavkama u osnovnoj FAI konfiguracijskoj skripti *fai.conf*, a sam *\$NFSROOT* direktorij rekonstruira se prema tim postavkama pozivanjem FAI-eve inijalizacijske skripte *fai-setup*.

Prilikom rekonstrukcije *\$NFSROOT* direktorija na FAI serveru u njemu se (između drugih manjih stvari) radi i sljedeće:

- Instaliraju se Debian paketi te razne konfiguracijske skripte potrebne za pravilan rad FAI mrežnog operacijskog sustava
- Otvara i konfigurira se osnovni root korisnički račun na FAI mrežnom operacijskom sustavu
- U njega se kopira osnovna FAI konfiguracijska datoteka *fai.conf*
- U njega se također kopiraju sve skripte koje čine kostur FAI sustava

Po ovom gore popisu jasno je nakon intervencija u kojim dijelovima FAI sustava je potrebno izazvati rekonstrukciju FAI *\$NFSROOT* direktorija.

5.5.1.5. Konfiguracija klijenata.

Do sad opisani dijelovi FAI sustava na serveru isti su na svim njegovim instalacijama. Ostali su još dijelovi u kojima se konfigurira što će se i točno kako će se instalirati na klijente u mreži koristeći FAI sustav. Ti se podaci nalaze na ukupno dva mjesta:

- U konfiguracijskoj datoteci za DHCP server */etc/dhcpd.conf*. Opis postavki koje se postavljaju u ovoj konfiguracijskoj datoteci bitnih za rad FAI servera opisan je malo kasnije u pododjeljku 5.7.1.
- U direktoriju na serveru koji ćemo u daljnjem dijelu ovog rada zvati *\$FAI*. Točna putanja ovog direktorija također se specificira u osnovnoj FAI konfiguracijskoj datoteci *fai.conf*, a po defaultu je */etc/local/share/fai*. Kasnije ćemo u pododjeljku 5.5.2 detaljno prikazati način korištenja ovog dijela konfiguracije tokom instalacije klijentovog lokalnog operacijskog sustava, a u odjeljku 5.7 kako se točno konfigurira jedna FAI instalacija klijenta na mreži.

5.5.2. Mrežni operacijski sustav

Nakon što se na klijentu podigne FAI mrežni operacijski sustav, na njemu se obavljaju sljedeće aktivnosti:

- Kao read-only root particija pomoću NFS protokola mounta se *\$NFSROOT* direktorij s FAI servera. Taj direktorij na FAI serveru mora biti specificiran na dva mjesta - u konfiguracijskim datotekama *dhcpd.conf* (vidi postavku option root-path) te *fai.conf* (vidi varijablu *\$NFSROOT*) kao osnovno (root) stablo za FAI mrežni operacijski sustav na klijentima. Dijelovi *\$NFSROOT* direktorija u koje sustav (bilo FAI sustav ili sam operacijski sustav) mora nešto privremeno zapisivati napravljeni su kao UNIX simbolične veze (symbolic link) na direktorije */tmp* direktorijskog stabla.
- Kao read-write particija na ramdisku klijenta mounta se */tmp* particija. Ramdisk particija je naziv za dio RAM memorije računala organizirane tako da joj se pristupa kroz neki datotečni sustav. */tmp* je jedina particija na koju (osim na klijentove lokalne hard diskove) FAI mrežni sustav zapisuje bilo kakve podatke tokom instalacije operacijskog sustava na klijenta. Time je osigurano da klijent ne može utjecati na podatke zapisane na FAI serveru.
- Kao read-only datotečni-sustavi pod direktorijem */tmp/target* mountaju se sve vidljive particije na lokalnim klijentovim diskovima. Samu detekciju lokanog hardwarea radi jezgra FAI mrežnog operacijskog sustava, koristeći programsku podršku ugrađenu u nju prilikom njenog stvaranja (za detaljniji opis jezgre FAI mrežnog operacijskog sustava vidi dodatak na kraju ovog rada) dok onda FAI u svojim skriptama prema tim podacima zna kako i koje particije može mountati s klijentovih lokalnih hard diskova.
- Kao direktorij specificiran u FAI konfiguracijskoj datoteci *fai.conf* (varijabla *\$FAI*) mounta se read-only *\$FAI* direktorij s FAI servera. To je način na koji FAI skripte koje se izvršavaju na klijentu saznaju točno kako i što treba instalirati na klijenta.
- Svi zapisi o događanjima (logovi) prilikom podizanja mrežnog operacijskog sustava te tokom same FAI instalacije zapisuju se na klijenta u direktorij */tmp/fai*. Točna putanja ovog direktorija također može biti promijenjena postavljanjem nove vrijednosti varijabli *\$LOGDIR* u osnovnoj FAI konfiguracijskoj datoteci *fai.conf*.

- Ukoliko je u konfiguracijskoj datoteci *dhcpcd.conf* u varijabli *\$FAI_FLAGS* za trenutnog klijenta upisana opcija *createvt* onda se na klijentu podižu ukupno tri konzolna terminala dostupna pritiskom na kombinacije tipaka *Ctrl-Alt-F1*, *Ctrl-Alt-F2* ili *Ctrl-Alt-F3*.
- Ukoliko je u konfiguracijskoj datoteci *dhcpcd.conf* u varijabli *\$FAI_FLAGS* za trenutnog klijenta upisana opcija *sshd* onda se na njemu pokreće SSH server program koji omogućuje pristup klijentu s drugih računala tokom izvođenja same instalacije operacijskog sustava.
- Na kraju uspješnog izvršavanja osnovnih FAI skripti svi logovi s klijenta prepisuju se na FAI server korištenjem korisničkog računa *\$LOGUSER* na FAI serveru. Taj račun je na FAI serveru otvorila i podesila osnovna instalacijska skripta FAI sustava *fai-setup*.

Nakon izvršavanja svih skripti FAI sustava na klijentu korisniku se ostavlja mogućnost da pritiskom na tipku RETURN pokrene ponovno podizanje računala ili da pritiskom kombinacije tipaka *Ctrl-C* pristupi osnovnom (root) korisničkom računu na klijentu. Ukoliko je u konfiguraciji na FAI serveru u konfiguracijskoj datoteci *dhcpcd.conf* za trenutnog klijenta u varijabli *\$FAI_FLAGS* zadana vrijednost *reboot* onda korisniku nije ponuđen ovaj izbor već po završetku izvršavanja skripti FAI sustava automatski dolazi do novog podizanja klijenta.

5.5.3. Povezanost FAI servera i klijenata

Za potpuno razumijevanje rada FAI sustava izrazito je bitno točno razumjeti sve dodirne točke između klijenta na kojem se izvršava FAI automatska mrežna instalacija i FAI servera. Prvo dodirno mjesto je inicijalni kontakt koji klijent ostvaruje s FAI serverom koristeći neki od prije spomenutih mrežnih protokola (BOOTP ili DHCP) te klijentovo dohvaćanje jezgre FAI mrežnog operacijskog sustava koristeći TFTP mrežni protokol. Dohvaćanje jezgre operacijskog sustava čini i sam kraj tog inicijalnog kontakta FAI servera i klijenta.

Drugi kontakt ostvaruje klijent po podizanju FAI mrežnog operacijskog sustava. To je mountanje svog osnovnog (root) datotečnog sustava s FAI servera koristeći NFS mrežni protokol. Sam osnovni (root) datotečni sustav nalazi se pripremljen na FAI serveru u direktoriju *\$NFSROOT*. Taj kontakt traje tokom cijelog postupka instalacije klijentovog lokalnog operacijskog sustava.

Treći kontakt čini klijentovo mountanje, također korištenjem NFS mrežnog protokola, *\$FAI* direktorija s FAI servera u kojem se nalaze konfiguracijske datoteke s uputama o tome kako treba instalirati novi klijentov lokalni operacijski sustav. Ovaj kontakt također traje tokom cijelog postupka instalacije klijentovog lokalnog operacijskog sustava.

Do četvrtog i posljednjeg kontakta FAI servera i klijenta dolazi na kraju same instalacije klijentovog lokalnog operacijskog sustava. Tada klijent pristupa FAI serveru kroz korisnički račun, posebno pripremljen tijekom instalacije FAI sustava na FAI serveru te na njega sprema sve prikupljene informacije (logove) o toku instalacije. Na ovaj način omogućen je uvid u te podatke radi eventualnih kasnijih analiza klijenta bez obzira na to je li klijent trenutno pristupačan ili ne.

Korisni detalj vezan uz rad FAI sustava je taj da je on zadužen samo za instalaciju operacijskog sustava na klijenta te ne propisuje niti forsira nikakva dodatna pravila kojih bi se klijent morao pridržavati tijekom svog rada. To znači da FAI server, osim za potrebe instalacije ili održavanje klijentovog lokalnog operacijskog sustava, uopće nije potreban, tj. da eventualna održavanja samog FAI servera neće ugroziti rad ostatka računalne mreže.

5.6. Instalacija FAI servera

5.6.1. Priprema za instalaciju

Prije same instalacije serverskog dijela FAI sustava, kao što je već ranije navedeno, potrebno je na računalo koje će služiti kao budući FAI server instalirati sljedeće pomoćne sustave:

- BOOTP ili DHCP server
- TFTP server
- NFS server

Općenito se njihovo instaliranje svodi samo na instaliranje odgovarajućih Debian paketa. Točan popis Debian paketa korištenih na našem razvojnom FAI serveru je sljedeći:

- *dhcp*
- *tftpd-hpa*
- *nfs-server*,

ali u detaljniji opis tih instalacija nećemo ulaziti u okvirima ovog rada.

5.6.2. Instalacija

5.6.2.1. Instalacija osnovnih Debian paketa

Sam FAI sustav dolazi zapakiran u dva osnovna Debian paketa:

- *fai*
- *fai-kernels*,

a koji prilikom svoje instalacije instaliraju još i sljedeće Debian pakete, naravno samo ukoliko se oni od prije već ne nalaze instalirani na tom računalu:

- *debootstrap*
- *netboot*
- *ssh-server*

Nakon postupka klasične mrežne Debian instalacije tih paketa na našem će se FAI serveru nalaziti svi dijelovi FAI sustava osim:

- konfiguracije instalacije pojedinih klijenata na mreži
- osnovnog (root) direktorija (*\$NFSROOT*) za klijente tijekom njihove instalacije.

Slijede kraći opisi gore navedenih Debian paketa te njihovog korištenja od strane FAI sustava.

5.6.2.1.1. *Debian paket fai*

Osnovni Debian paket koji sadrži FAI sustav.

5.6.2.1.2. *Debian paket fai-kernels*

Debian paket *fai-kernels* sadrži u sebi nekoliko unaprijed pripremljenih jezgara Debian GNU/Linux operacijskog sustava koje se mogu koristiti za podizanje FAI mrežnog operacijskog sustava. Ovaj nam paket nije potreban na serveru ukoliko ćemo te jezgre sami pripremati.

Samo pripremanje jezgre FAI mrežnog operacijskog sustava te s tim i `$NFSROOT` direktorija bit će opisano malo kasnije.

5.6.2.1.3. *Debian paket debootstrap*

Debian paket *debootstrap* koji sadrži programsku podršku za instaliranje osnovnog dijela Debian GNU/Linux operacijskog sustava, bez potrebe za dodatnim alatima za pristupanje arhivi Debian GNU/Linux distribucije. FAI ovaj alat koristi za pripremanje svog `$NFSROOT` direktorija.

5.6.2.1.4. *Debian paket netboot*

Debian paket *netboot* koji sadrži alate potrebne za pripremanje jezgre Debian GNU/Linux operacijskog sustava kako bi se taj operacijski sustav mogao podići preko mreže. To pripremanje uključuje podešavanje jezgre operacijskog sustava da svoj osnovni (root) datotečni sustav pokuša dohvatiti preko mreže koristeći NFS protokol.

Točan naziv programa koji FAI sustav koristi iz ovog paketa je *mknbi-linux*.

Osim netboot sustava za podizanje mrežnih operacijskih sustava također je moguće koristiti etherboot sustav. Oba su sustava razvijena metodom otvorenog izvornog koda, a njihov izvorni kod te sve detaljnije informacije o njima dostupne su na WWW adresama: <http://netboot.sourceforge.com> i <http://etherboot.sourceforge.com>.

5.6.2.1.5. *Debian paket ssh-server*

Debian paket koji sadrži programsku podršku koja omogućuje spajanje na računalo s drugih računala na mreži koristeći SSH protokol. Taj protokol omogućuje sigurnu enkriptiranu mrežnu komunikaciju među računalima, a FAI sustav ga koristi za sigurnu komunikaciju između FAI servera i klijenata.

Umjesto SSH protokola, FAI sustav se može konfigurirati (u svojoj osnovnoj konfiguracijskoj datoteci *fai.conf*) da koristi RSH protokol, ali korištenje tog protokola se ne preporuča zbog nesigurnog (neenkriptiranog) prijenosa informacija preko mreže.

5.6.2.2. **Prepravljanja FAI sustava**

Nakon instalacije Debian paketa potrebnih za rad FAI sustava trenutak je za bilo kakve izmjene u samoj strukturi FAI sustava. Prilikom prvog upoznavanja s FAI sustavom najvjerojatnije nema potrebe za ovakvim korakom, ali s obzirom da je cijeli FAI sustav projekt na kojem aktivno rade stotine ljudi duž cijelog svijeta, svakim danom se na internetu mogu naći novi dodaci i popravci za taj sustav koje onda osoba zadužena za FAI sustav može poželjeti u sklopu svog sustava.

Tokom našeg rada na FAI sustavu također smo napravili niz dorada na samom sustavu. Točni opisi te izvorni kod ovih dorada može se naći na CD-u priloženom uz ovaj rad.

5.6.2.3. Osnovno konfiguriranje FAI sustava

Idući korak je osnovno konfiguriranje FAI sustava. To se konfiguriranje obavlja postavljanjem osnovnih postavki za rad FAI sustava u osnovnoj konfiguracijskoj skripti *fai.conf*. Ovdje ćemo dati pregled postavki iz konfiguracijske datoteke te njihove vrijednosti kako su bile postavljene na našoj razvojnoj implementaciji FAI servera.

Osim dolje navedenih postavki postoje još neke koje nije predviđeno mijenjati, kao što je točna putanja *\$NFSROOT* direktorija na FAI serveru.

5.6.2.3.1. *\$ftpserver*

\$ftpserver je varijabla u kojoj se upisuje naziv ili IP adresa servera na mreži koji sadrži arhivu Debian GNU/Linux distribucije, a koja će biti korištena prilikom FAI automatske mrežne instalacije operacijskih sustava na klijentima.

Moguće vrijednosti ove varijable su ftp.debian.org ili ftp.hr.debian.org, a za našu instalaciju FAI sustava napravili smo zrcalnu sliku Debian GNU/Linux distribucije na računalu s nazivom *fourier.math.hr* pa je vrijednost varijable *\$ftpserver* postavljena baš na naziv tog računala.

5.6.2.3.2. *\$debdist*

Varijabla *\$debdist* sadrži naziv verzije Debian GNU/Linux distribucije koja se koristi pri radu FAI sustava. Kod nas je korištena verzija *woody* koja je u trenutku pisanja ovog rada i trenutno zadnja stabilna verzija Debian GNU/Linux distribucije. Prethodne dvije verzije ove Linux distribucije zvale su se *sid* i *potato*.

5.6.2.3.3. *\$FAI_DEBOOTSTRAP*

U varijablu *\$FAI_DEBOOTSTRAP* zapisuje se točan direktorij u kojem se na računalu danom *\$ftpserver* varijablom nalazi osnovni dio Debian GNU/Linux distribucije te točan protokol kojim se ti podaci dohvaćaju sa servera. Standardno se Debian GNU/Linux distribucija nalazi u direktoriju *debian* pa je dako bilo i na našem mirroru te distribucije. Mogući protokoli su FTP i HTTP, a u našoj implementaciji FAI sustava korišten je HTTP.

5.6.2.3.4. *\$FAI_SOURCES_LIST*

U varijabli *\$FAI_SOURCES_LIST* popisana su eventualna dodatna mjesta odakle FAI sustav može u toku svoje instalacije te instalacije klijenata dohvaćati dodatne Debian pakete, npr. računalo s nazivom *security.debian.org* gdje se stavljaju sve eventualne hitne popravke za Debian GNU/Linux distribuciju.

5.6.2.3.5. *\$NFSROOT_PACKAGES*

U varijabli *\$NFSROOT_PACKAGES* nalazi se popis dodatnih Debian paketa koji trebaju biti instalirani na FAI mrežnom operacijskom sustavu – dakle u *\$NFSROOT* direktoriju na FAI serveru. Primjer ovakvog Debian paketa je *ssh*

koji omogućava FAI sustavu da koristi SSH mrežni protokol za komunikaciju između FAI servera i klijenta tokom samog postupka instalacije klijenta.

5.6.2.3.6. *\$FAI_ROOTPW*

U varijabli *\$FAI_ROOTPW* postavlja se enkriptirana lozinka za root korisnički račun na FAI mrežnom operacijskom sustavu. Samo enkriptiranje lozinke provodi se korištenjem UNIX programa *mkpasswd*.

5.6.2.3.7. *\$SSH_IDENTITY*

U varijabli *\$SSH_IDENTITY* moguće se zapisati direktorij na FAI serveru koji sadrži neke javne ključeve. Korisničkom računu identificiranom tim ključevima bit će onda dozvoljeno spajanje na klijente bez provjere lozinke dok je na klijentu podignut FAI mrežni operacijski sustav.

5.6.2.3.8. *\$KERNELPACKAGE*

U varijabli *\$KERNELPACKAGE* upisuje se puni naziv lokalno spremljenog Debian paketa koji sadrži jezgru operacijskog sustava korištenu za FAI mrežni operacijski sustav na klijentima. Također mora biti zadana puta putanja do direktorija na FAI serveru u kojem se nalazi taj Debian paket.

5.6.2.3.9. *\$LOGUSER*

U varijabli omjer zadan je naziv korisničkog računa na FAI serveru kojeg će FAI sustav koristiti za komunikaciju između FAI servera i klijenata te u čijem će osnovnom (home) direktoriju biti spremljeni svi izvještaji o toku provedenih FAI automatskih mrežnih instalacija na klijentima.

5.6.2.3.10. *\$FAI_REMOTESH* i *\$FAI_REMOTECP*

Varijablama *\$FAI_REMOTESH* i *\$FAI_REMOTECP* zadaju se nazivi programa korištenih za komunikaciju između FAI servera i klijenta tijekom FAI automatske mrežne instalacije operacijskog sustava. U našoj implemetaciji FAI sustava korišteni su programi *ssh* i *scp* koji koriste SSH mrežni protokol. Odgovarajući programi za korištenje RSH mrežnih protokola bili bi *rsh* i *rcp*.

5.6.2.4. Inicijaliziranje FAI sustava, kreiranje *\$NFSROOT* direktorija

Idući korak u instalaciji FAI servera je inicijalizacija samog FAI sustava. To se obavlja pokretanjem Linux shell skripte FAI sustava */usr/sbin/fai-setup*. Sama inicijalizacija odrađuje se prema postavkama iz osnovne FAI konfiguracijske datoteke *fai.conf*.

Po završetku izvršavanja ove dugotrajne skripte na FAI serveru bit će napravljene sljedeće stvari:

- Otvara se i konfigurira *\$LOGUSER* korisnički račun. Na njega nije moguć pristup osim sa klijenata tijekom postupka FAI automatske mrežne instalacije, a njegova uloga u sklopu FAI sustava već je ranije opisana u pododjeljku 5.6.2.3.9.
- Konfigurira se NFS server tako da su klijentima na mreži, korištenjem NFS mrežnog protokola, dostupni direktoriji potrebni za ispravan rad FAI sustava. Savjetujem, po završetku izvršavanja skripte *fai-setup*,

osobnu provjeru datoteke `/etc/exports` u kojoj je ova konfiguracija upisana kako bi se izbjegle neke kasnije teško ulovljive pogreške. Primjer takve je višestruko konfiguriranje istih direktorija za korištenje kroz NFS mrežni protokol što može dovesti no neočekivanog ponašanja samog NFS server programa.

- Kreira se `$NFSROOT` direktorij sa svom potrebnom programskom podrškom za odabrani FAI mrežni operacijski sustav. Između ostalog na njemu se otvara i priprema osnovni (root) korisnički račun.

Kasnije prilikom bilo kakve promjene osnovne FAI konfiguracijske datoteke `fai.conf` (npr. promjena verzije FAI mrežnog operacijskog sustava ili dodavanje novih paketa u taj operacijski sustav) potrebno je ponovo rekreirati `$NFSROOT` direktorij na FAI serveru pozivanjem ove iste skripte `fai-setup`. Pozivanje te skripte neće izazvati nikakve štetne posljedice jer se unutar nje za pojedine korake, kao npr. postavljanje `$LOGUSER` korisničkog računa, provjerava jesu li suvišni te ako jesu onda ih se preskače. Drugim riječima ponovno pozivanje ove skripte neće imati za posljedicu otvaranje višestrukih `$LOGUSER` korisničkih računa ili višestruko konfiguriranje NFS servera.

Izrazito je važno istaknuti da sve promjene u `$NFSROOT` direktoriju moraju biti generirane ovom skriptom, a nikako nije dozvoljeno te promjene ručno provoditi. To je zato da bi se `$NFSROOT` direktorij mogao drugi puta, u slučaju potrebe, automatizirano regenerirati bez gubljenja promjena napravljenih u njemu.

5.7. Konfiguracija mreže FAI klijenata

5.7.1. Osnovna konfiguracija mreže klijenata - DHCP

Sljedeći korak u instalaciji FAI sustava je konfiguriranje podataka koje klijenti dobivaju od FAI servera prilikom svog buđenja korištenjem BOOTP ili DHCP mrežnog protokola.

U našoj implementaciji FAI sustava koristili smo DHCP mrežni protokol pa će ovdje biti opisana njegova konkretna upotreba. Razlika između ta dva protokola je u mogućnosti DHCP mrežnog protokola da klijentu dodjeli, umjesto fiksne, slučajnu mrežnu IP adresu iz svoje 'zalihe' slobodnih IP adresa. Ta mogućnost DHCP mrežnog protokola neće biti opisivana niti korištena u sklopu ovog rada, pa za naše potrebe izbor između DHCP i BOOTP mrežnih protokola pretstavlja jedino sintaksnu razliku u nazivu nekih konfiguracijskih datoteka i programa te sintaksi kojom se zadaje sama konfiguracija korištenja mrežnog protokola. Više detalja o korištenju BOOTP protokola može se naći u službenoj dokumentaciji FAI sustava.

Osnovna i jedina konfiguracijska datoteka za DHCP mrežni protokol u kojoj mi postavljamo postavke bitne za rad FAI sustava je `/etc/dhcpd.conf`. Važno je napomenuti da promjene napravljene u toj konfiguracijskoj datoteci postaju važeće tek ponovnim pokretanjem DHCP server programa (daemon) na FAI serveru. To pokretanje DHCP servera ostvaruje se naredbom:

```
/etc/init.d/dhcp restart
```

ili

```
/etc/init.d/dhcp force-reload
```

U daljnjem dijelu ovog pododjeljka navedene su i obajašnjene samo postavke iz *dhcpcd.conf* konfiguracijske datoteke bitne za rad FAI sustava dok je do potpunijih informacija o radu DHCP mrežnog protokola te o konfiguriranju DHCP servera moguće pronaći na Linux *man* stranicama: *dhcpcd(8)*, *dhcpcd.conf(5)* i *dhcp-options(5)*.

5.7.1.1. server-name

Postavkom *server-name* određuje se naziv računala koje služi kao DHCP server (dakle i kao FAI server) kako će biti proslijeđeno klijentima na mreži.

Primjer:

```
server-name "cauchy.math.hr";
```

5.7.1.2. host-name

Postavkom *host-name* određuje se naziv klijenta koji će biti proslijeđen klijentu. Ukoliko ovaj podatak nije zadan klijentu će kao njegovo ime biti proslijeđena njegova IP adresa (vidi 5.7.1.6).

Primjer:

```
host-name "gauss";
```

5.7.1.3. next-server

Postavkom *next-server* moguće je, ukoliko se FAI server ne koristi i kao Install server, zadati adresu Install servera.

Primjer:

```
next-server 192.84.105.155;
```

5.7.1.4. option domain-name

Postavkom *option domain-name* određuje se naziv internet DNS domene kojoj pripada klijent.

Primjer:

```
option domain-name "math.hr";
```

5.7.1.5. hardware ethernet

Postavkom *hardware ethernet* zadaje se identifikacija mrežne kartice (MAC adresa – Median Access Control address) u klijentu. To je podatak po kojemu DHCP server prepoznaje i razlikuje pojedine klijente.

Primjer:

```
hardware ethernet 00:04:75:ca:fc:f1;
```

5.7.1.6. fixed-address

Postavkom *fixed-address* zadaje se fiksna IP adresa pridružena pojedinom klijentu.

Primjer:

```
fixed-address 192.84.105.167
```

5.7.1.7. filename

Postavkom *filename* zadaje se naziv jezgre Linux operacijskog sustava koja će se koristiti kao FAI mrežni operacijski sustav na klijentu.

Primjer:

```
filename "/boot/fai/vmlinuz-2.4.18df5"
```

5.7.1.8. option root-path

Postavkom *option root-path* klijentima se dostavlja puna putanja do pripremljenog *\$NFSROOT* direktorija na FAI serveru. Ova postavka mora biti usklađena s odgovarajućom postavkom u osnovnoj konfiguracijskoj datoteci FAI sustava *fai.conf*.

Primjer:

```
option root-path "/usr/lib/fai/nfsroot"
```

5.7.1.9. option option-170 (\$FAI_LOCATION)

Postavka *option option-170* ili *\$FAI_LOCATION* je prva od posebnih postavki potrebnih isključivo za rad FAI sustava te koje van tog sustava nemaju definiran značaj. Njome se klijentu prosljeđuje točna putanja *\$FAI* direktorija na FAI serveru u kojem se nalazi opis same instalacije koju će biti potrebno pronesti na klijentu.

Primjer:

```
option option-170 "192.84.105.159:/usr/local/share/fai"
```

5.7.1.10. option option-171 (\$FAI_ACTION)

Postavkom *option option-171* ili *\$FAI_ACTION* FAI server klijentu prosljeđuje uputu o osnovnoj akciji koju je potrebno provesti na tom klijentu prilikom njegovog idućeg podizanja FAI mrežnog operacijskog sustava.

Moguće vrijednosti ove postavke su:

- **sysinfo** – prilikom idućeg podizanja FAI mrežnog operacijskog sustava na klijentu neće doći do automatske mrežne instalacije lokalnog operacijskog sustava, već će se FAI administratoru omogućiti pristup do klijenta zbog eventualnog administriranja.
- **install** – prilikom idućeg podizanja FAI mrežnog operacijskog sustava na klijentu se pokreće FAI automatska mrežna instalacija operacijskog sustava. Točne upute o tome što i kako se instalira na pojedinog klijenta nalaze se navedene u FAI konfiguracijskim skriptama u *\$FAI* direktoriju te nekim početnim parametrima za klijenta zadanim u ovoj konfiguracijskoj skripti *dhcpd.conf*.
- **backup** – planirana, ali do ovog trenutka još uvijek ne implementirana akcija u sklopu FAI sustava. Trebala bi pokretati postupak backupiranja podataka s klijenta.

Osim gore navedenih predefiniranih akcija koje FAI sustav može odraditi moguće je dodati nove posebno prilagođene akcije sustavom 'hook' skripti ugrađenim u FAI sustav. Osnova tog postupka je dodavanje, pod nazivom akcije koju želimo ugraditi u FAI sustav, nove skripte u direktorij *\$FAI/hook* na FAI serveru. Takva skripta zove se 'hook skripta', a bit će automatski pokrenuta prilikom idućeg podizanja FAI mrežnog operacijskog sustava na

klijentu ukoliko postavkom `$FAI_ACTION` zatraži odgovarajuća novododana akcija.

Detaljniji opis hook mehanizma u FAI sustavu prelazi izvan okvira ovog rada, a više informacija te detaljni primjeri mogu se naći u sklopu standardne FAI distribucije.

Primjer korištenja ove postavke:

```
option option-171 "sysinfo";
```

5.7.1.11. option option-172 (\$FAI_FLAGS)

Postavkom `option option-17` ili `$FAI_FLAGS` zadaju se dodatni parametri za rad FAI sustava na klijentu.

Primjer korištenja:

```
option option-172 "verbose createvt sshd";
```

Slijedi točan opis svih parametara koje je na ovom mjestu moguće postaviti:

5.7.1.11.1. verbose

Parametar *verbose* izaziva detaljnije zapisivanje informacija o toku FAI automatizirane mrežne instalacije na klijentu. Izrazito koristan prilikom traženja eventualnih anomalija u instalaciji pojedinog klijenta te ga je preporučljivo uvijek ostaviti postavljenog.

5.7.1.11.2. debug

Parametar *debug* izaziva podizanje FAI mrežnog operacijskog sustava u debug modu. To ima kao posljedice:

- dodatne ispise prilikom izvođenja FAI skripti na klijentu
- dodatne upite korisniku tokom izvršavanja mrežne instalacije operacijskog sustava na klijentu. Ovo dakako ne možemo onda nazvati automatskom instalacijom.

5.7.1.11.3. reboot

Parametar *reboot* izaziva automatsko rebootanje klijenta po završetku izvođenja FAI skripti na klijentu. Ukoliko ovaj parametar nije postavljen, FAI sustav po završetku svog posla čeka da korisnik ručno pokrene rebootanje klijenta ili pritiskom kombinacije tipaka *Ctrl-C* uđe u osnovni (root) shell FAI mrežnog operacijskog sustava.

5.7.1.11.4. createvt

Parametar *createvt* upućuje FAI sustav da po dizanju FAI mrežnog operacijskog sustava na klijentu napravi dva dodatna virtualna konzolna terminala. Sami terminali dostupni su sa klijentove konzole pritiskom kombinacija tipaka *Ctrl+Alt+F1*, *Ctrl+Alt+F2* ili *Ctrl+Alt+F3*.

5.7.1.11.5. sshd

Parametar *sshd* upućuje FAI sustav da po dizanju FAI mrežnog operacijskog sustava na klijentu pokrene i server program (daemon) za SSH

protokol. Time se dozvoljava pristup sa mreže na klijenta za vrijeme trajanja same automatizirane instalacije njegovog lokalnog operacijskog sustava.

Kako bi korištenje ovog parametra uopće imao ikakvog učinka, prilikom kreiranja *\$NFSROOT* direktorija na FAI serveru potrebno je u njega instalirati Debian paket s podrškom za SSH protokol, tj. potrebno je prije kreiranja *\$NFSROOT* direktorija u osnovnoj FAI konfiguracijskog datoteci *fai.conf* navesti *ssh* Debian paket u popisu dodatnih Debian paketa za *\$NFSROOT* direktorij.

Primjer bitnih dijelova jedne *dhcpcd.conf* konfiguracijske datoteke

```
default-lease-time 600;
max-lease-time 7200;
use-host-decl-names on;
server-name "cauchy.math.hr";
# next-server 192.84.105.155;
option subnet-mask 255.255.255.0;
option broadcast-address 192.84.105.255;
option routers 192.84.105.1;
option domain-name-servers 192.84.105.1;
option domain-name "math.hr";
option ntp-servers 192.84.105.151;

subnet 192.84.105.0 netmask 255.255.255.0
{
    # -----
    # FAI clients.
    # -----

    group
    {
        option root-path "/usr/lib/fai/nfsroot";

        option option-170 "192.84.105.159:/usr/local/share/fai"; # FAI_LOCATION
        option option-171 "sysinfo"; # FAI_ACTION
        option option-172 "verbose createvt sshd"; # FAI_FLAGS

        filename "/boot/fai/vmlinuz-2.4.18df5";

        host gauss
        {
            hardware ethernet 00:00:f8:03:df:c9;
            fixed-address 192.84.105.147;
            # option option-171 "install";
        }
    }
}
```

5.7.2. Konfiguracija FAI automatske mrežne instalacije

Cijeli posao koji FAI sustav obavlja na klijentu bazira se na sustavu klasa. Prva stvar koju određuje je točan popis klasa kojima pripada pojedini klijent. Na taj način omogućeno nam je jednostavno grupiranje srodnih dijelova različitih instalacija te posve uklanjanje potrebe za dupliciranjem posla prilikom instalacija operacijskih sustava na našim klijentima.

Primjer korištenja FAI klasa:

Prilikom instalacije programske podrške za NTP mrežni protokol na klijenta (mrežni protokol za usklađivanje vremena na računalima u mreži) potrebno je na njega instalirati neke Debian pakete. FAI sustav nam omogućava da samo kažemo: ‘Na računala X, Y i Z želim instalirati podršku za NTP’, a da opis te instalacije zapišemo na samo jedno mjesto – u opis

vezan uz odgovarajuću klasu NTP, iako se instalacije operacijskih sustava na računalima X, Y i Z mogu razlikovati u drugim dijelovima.

5.7.2.1. Određivanje klasa kojima pripada klijent

Neke od klasa pridružuju se automatski svakom klijentu u samom kosturu FAI sustava. To su klase:

- DEFAULT
- \$HOSTNAME
- LAST

Klasa DEFAULT uvijek je dodana na početak popisa klasa kojima pripada klijent, dok su klase \$HOSTNAME i LAST dodane tim redoslijedom na kraj tog popisa.

Ostale klase pridružuju se klijentu skriptama koje se nalaze u direktoriju *\$FAI/class*, a čiji nazivi odgovaraju sljedećem regularnom izrazu: *S[0-9]*.{pl,sh,source}*. Te skripte izvršavaju se abecednim redom, a klase kojima pripada klijent definiraju se na sljedeći način:

- skripte čiji naziv završava na *.pl* ili *.sh* samo na standardni izlaz ispišu imena klasa odvojena razmacima ili oznakama za kraj reda.
- skripte čiji naziv završava na *.source* zapišu klase u varijablu *\$newclasses*. Pri tome se mora paziti da se ne koriste varijable koje već interno koriste skripte iz kostura FAI sustava, kao npr. *\$classes*.

Klase moraju biti definirane redom od generalnijih prema specifičnijima. Pojedini klijenti unutar skripti razlikuju se prema vrijednosti varijable *\$HOSTNAME*, a njenu je vrijednost klijent dobio od DHCP servera (vidi ranije opis korištenja postavke *host-name* u konfiguracijskoj datoteci *dhcpcd.conf*).

5.7.2.2. Postavljanje dodatnih varijabli

Sljedeći korak u konfiguraciji instalacije klijenta je postavljanje dodatnih varijabli potrebnih za kasnije detaljno konfiguriranje instaliranog operacijskog sustava na klijentu. Opis načina korištenja tih varijabli bit će dan u pododjeljku 5.7.2.5, a ovdje ćemo samo navesti način na koji se oni definiraju te primjere uloga koje one imaju u našoj implementaciji FAI sustava.

Same varijable definiraju se Linux shell skriptama u direktoriju *\$FAI/class* nazvanim imenima klase kojoj pripada klijent, a s nastavkom *.var*.

Neki primjeri korištenja ovakvih varijabli su sljedeći:

- *\$FAI_KEYMAP* – varijabla čija vrijednost određuje raspored tipki na tipkovnici koji će biti podržan na klijentovom lokalnom operacijskom sustavu
- *\$rootpw* – enkriptirana lozinka za osnovni (root) korisnički račun na klijentovom lokalnom operacijskom sustavu
- *\$addpackages* – varijabla s popisom Debian paketa koje je potrebno dodatno instalirati na klijenta, ali ne iz korištene Debian GNU/Linux arhive već lokalno spremljenih na serveru u direktoriju *\$FAI/files/packages*

5.7.2.3. Particioniranje i formatiziranje lokalnih diskova na klijentu

U ovom koraku konfigurira se način na koji FAI sustav prilikom svoje automatske instalacije klijetovog operacijskog sustava particionira i formatizira lokalne klijentove diskove. Upute za taj postupak upisuju se u konfiguracijske datoteke koje se nalaze u direktoriju *\$FAI/disk_config*, a zovu prema imenima klasa kojima pripada klijent.

Točan format zapisa ovih konfiguracijskih datoteka za particioniranje i formatiziranje klijentovih lokalnih diskova prelazi okvire ovog rada, se može naći detaljno opisan i objašnjen u dodatku na kraju ovog rada.

Sve eventualne pogreške i nelogičnosti u korištenim konfiguracijskim datotekama FAI sustav će promptno detektirati i prijaviti. Primjer jedne takve pogreške je klijentovo pripadanje dvjema različitim klasama od kojih svaka definira svoj, različit način particioniranja istog diska.

Po završetku postupka particioniranja i formatiziranja klijentovih lokalnih diskova FAI sustav će na klijentovu osnovnu (root) particiju zapisati u datoteku */etc/fstab* točan popis i opis formatiziranih particija.

5.7.2.4. Instaliranje Debian paketa na klijenta

Idući korak je konfiguriranje Debian paketa koji trebaju biti instalirani na klijentov lokalni operacijski sustav. To se obavlja popisivanjem tih paketa u konfiguracijskim datotekama na FAI serveru u direktoriju *\$FAI/package_config* nazvanim po imenima klasa kojima pripada klijent.

Točan format zapisa ovih konfiguracijskih datoteka prelazi okvire ovog rada, ali se može naći u dodatnoj dokumentaciji na priloženom CD-u.

U ovim konfiguracijskim datotekama moguće je definirati koji Debian paketi će biti instalirani na klijentov lokani operacijski sustav, ali i obrnuto: koji Debian paketi trebaju biti uklonjeni s tog sustava. Korištenjem ove tehnike moguće je postići da jedna specijaliziranija klasa ukloni sa klijentovog operacijskog sustava Debian pakete koje je na njega instalirala neka općenitija klasa. Za primjer možemo navesti programsku podršku potrebnu za korištenje različitih Linux boot loadera. Inicijalnom klasom DEFAULT na klijentov lokalni operacijsku sustav instalira se podrška za LILO boot loader, ali ukoliko želimo koristiti npr. GRUB boot loader onda klasa GRUB uklanja te zamjenjuje već instaliranu programsku podršku onom koja je potrebna za rad GRUB boot loadera.

Gore navedeni mehanizam je također jedan od razloga zašto se klase kojima pripada klijent moraju navoditi od općenitijih prema specifičnijima.

5.7.2.5. Konfiguriranje klijentovog operacijskog sustava

Posljednji dio konfiguracije FAI sustava koji moramo napraviti su skripte koje se pokreću na klijentu nakon što su na njemu particionirani i formatizirani diskovi te instalirani svi gore navedeni Debian paketi. Zadatak tih skripti je fino podesiti klijentov lokalni operacijski sustav.

Skripte za fino podešavanje klijentovog lokalnog operacijskog sustava nalaze se u konfiguracijskom direktoriju *\$FAI/scripts*, a mogu biti zadane na dva načina:

- Prvi način je nazivom klase kojoj pripada klijent.

- Drugi način je nazivom koji zadovoljava regularni izraz 'S[0-9]*', a nalazi se u direktoriju nazvanom po imenu klase kojoj pripada klijent. Na ovaj način omogućeno je da pripadanje klijenata jednoj klasi izazove pokretanje više od jedne skripte.

Same skripte izvršavaju se redom kojim su definirane klase kojima pripada klijent, a skripte unutar direktorija nazvanog po imenu klase abecednim redom.

Primjeri nekih poslova koje mogu odrađivati ove skripte:

- postavljanje korisničkih računa na klijentovom operacijskom sustavu
- popisivanje kernel modula koji moraju biti učitani prilikom podizanja novog klijentovog lokalnog operacijskog sustava u datoteku */etc/modules*
- konfiguriranje klijentovih mrežnih sučelja
- konfiguriranje pojedinih programskih paketa instaliranih na klijenta
- postavljanje sistemskog vremena na klijentu korištenjem NTP mrežnog protokola
- postavljanje prava pristupa na posebne direktorije na klijentu - */tmp*, */dev/fd0* i sl.

Ukoliko ove skripte trebaju na klijenta nasnimiti neke unaprijed pripremljene datoteke one moraju biti smještene u direktorij *\$FAI/files*. Tamo su te datoteke spremljene u direktoriju nazvanom po imenu datoteke, a unutar njega u datoteci nazvanoj po imenu klase. Za samo kopiranje ovih datoteka koriste se pomoćne FAI skripte *fcopy* i *ftar*. Za više detaljnijih informacija o ovim skriptama vidi *man* stranice *fcopy(8)* i *ftar(8)*.

Primjer:

Ako želimo na klijenta koji pripada klasi *KNOW_HOSTS* nasnimiti posebno pripremljenu */etc/hosts* datoteku onda bi: napravili slijedeće:

- U skriptu *\$FAI/scripts/KNOW_HOSTS* upisali naredbu *fcopy /etc/hosts*.
- U direktorij *\$FAI/files/etc/hosts* pod nazivom *KNOW_HOSTS* nasnimili željeni sadržaj */etc/hosts* datoteke.

5.8. Daljnji razvoj FAI sustava

Ovime smo opisali sve dijelove FAI sustava potrebne za njegovu primjenu na problem automatske instalacije operacijskih sustava te administriranja mreže računala.

FAI sustav zamišljen je i napravljen tako da bude lagan za nadograđivanje. U ovom radu opisana je konkretna primjena FAI sustava za administriranje mreže računala s *i386* arhitekturom koji rade pod Debian GNU/Linux operacijskim sustavom, ali ljudi diljem cijelog svijeta rade na razvoju ovog sustava pa se već danas mogu pronaći njegove inačice primjenjene na administriranje velikih mreža računala različitih platformi koje rade pod nizom drugih operacijskih sustava.

6. Zaključak

Prihvaćenost FAI sustava u sklop službene stabilne Debian GNU/Linux distribucije još je jedna činjenica koja nam govori o općenitoj potrebi za ovakvim sustavom.

Precizno definiranje potrebnih koraka te automatizacija njihovog provođenja u zadatku administracije računalnih infrastruktura prirodna je iduća stepenica u evoluciji. Ovakav razvoj nije slučajan niti neviđen. Primjer ovakvog razvoja može se naći npr. u automobilske industriji koja ga je doživjela krajem prve polovice prošlog stoljeća te je time doživjela svoj veliki procvat. Potpuna automatizacija u njihovim industrijskim procesima postigla je efekte analogne onima kakve sada vidimo u informatici. Materijalne, financijske i vremenske koristi koje omogućuje ovakva organizacija neprocijenjive su vrijednosti:

- Cijeli sustav može raditi većom brzinom i s puno većom sigurnošću.
- Povećana je kontrola nad sustavom te smanjen utjecaj ljudskih grešaka na rad cijelog sustava.
- Troškovi administracije sustava drastično su smanjeni. Neki dijelovi administracije, koji su tradicionalno bili skupi, postali su gotovo besplatni.
- Smanjeno je opterećenje na poslu administratora sustava. Njegova stručnost i dalje je potrebna, ali je puno bolje iskorištena.
- Moral ljudi koji obavljaju ovakav posao porastao je jer su iz standardnog opisa posla ovime izbačene mnoge rutinske i monotone dužnosti, a omogućene su puno veće prilike za kreativnost i razvoj vlastitih sposobnosti.

Izrazita ljepota i praktičnost ovakvog sustava može se vidjeti i u našem FAI sustavu. U njemu svi podaci potrebni za konfiguriranje jedne velike računalne infrastrukture sastavljene od više stotina računala mogu stati na jednu disketu. Svi ostali dijelovi FAI sustava su fiksni i ne razlikuju se. Ovisno o mreži računala koju želimo administrirati – same skripte koje čine kostur FAI sustava su jednake, paketi potrebni za instalaciju sve potrebne programske podrške lako su dostupni na internetu, a osnovne jezgre operacijskih sustava pripremljene za podizanje FAI mrežnog operacijskog sustava beznačajne su veličine. Ovime je cijena osiguravanja organizacije jedne velike infrastrukture smanjena na cijenu nekoliko disketa, a postignuta sigurnost daleko veća od one koju bi smo dobili povećavanjem stogosti pravila kojih se moraju pridržavati ljudi koji koriste sustav.

Do ovog koraka u razvoju, dakako, nije došlo trenutačno. Viziju budućnosti ljudi su imali već osamdesetih godina kada su velike kompanije poput Digitala ili IBMa počele proučavati i stvarati prve sustave za administraciju i instalaciju operacijskih sustava preko mreže. Tek razvojem moderne mrežne tehnologije ovakvi su se sustavi našli u mogućnosti funkcionirati na velikim infrastrukturama.

Danas je sve spremno. Imamo razvijenu mrežnu tehnologiju, imamo razvijene programske alate, imamo iza sebe mnoštvo iskustava iz različitih grana znanosti i industrije iz kojih možemo učiti, tako da očekujem da će sustavi kakve smo opisivali u ovom radu postati sve veći dio naše

svakodnevice. Koristi nastale njihovom primjenom učinit će život i rad ljudi bržim, korisnijim i ugodnijim.

7. Dodaci

7.1. Mrežno podizanje Debian GNU/Linux operacijskog sustava

7.1.1. *Bootstrap kod*

Inicijalni programski kod koji se izvršava na klijentu, a koji ima za dužnost pokrenuti klijentov operacijski sustav, nazivamo 'bootstrap kod'. Za računala s novijim mrežnim karticama on se nalazi već ugrađen u samoj mrežnoj kartici, dok ga se za starija računala može pripremiti i na nekom drugom mediju, npr. CD-u ili disketi.

Bootstrap kod koji se koristi na klijentu mora točno odgovarati specifikacijama mrežne kartice ugrađene u klijentu.

Prilikom istraživanja mi smo isprobali tri varijante ovakvog bootstrap koda:

7.1.1.1. Etherboot i Netboot bootstrap kod

Etherboot varijanta može se za razne mrežne kartice te s raznim dodatnim mogućnostima pronaći na web stranici <http://www.rom-o-matic.net>, dok se Netboot varijanta može napraviti korištenjem programa *makerom* koji se instalira u sklopu *netboot* Debian paketa.

Korištenje ovih varijanti bootstrap koda zahtjeva posebno pripremanje jezgre mrežnog operacijskog sustava programom *mknbi-linux* koji za Etherboot dolazi u sklopu Debian paketa *mknbi*, a za Netboot u sklopu Debian paketa *netboot*.

Osim toga, ukoliko se u klijentu nalazi 3com mrežna kartica onda je potrebno tu istu obrađenu jezgru mrežnog operacijskog sustava još naknadno pripremiti 3com-ovim programom *imggen*. Pri tom treba paziti na odabir međusobno kompatibilnih verzije *imggen* i *mknbi-linux* programa.

7.1.1.2. PXE

Kreiranje PXE varijante bootstrap koda nismo proučavali, već smo koristili kod koji se već nalazi u korištenim 3com mrežnim karticama.

7.1.2. *Opis postupka podizanja mrežnog operacijskog sustava*

7.1.2.1. Netboot/Etherboot bootstrap kod

- 1) Bootstrap kod na klijentu šalje na mrežu DHCP zahtjev.
- 2) DHCP server program na FAI serveru prepoznaje klijenta po njegovoj MAC adresi te mu vraća osnovne informacije o njemu (IP adresu, naziv, informacije o osnovnim dostupnim serverima na mreži i sl.) uključivo s nazivom datoteke koju klijent može dohvatiti na mreži, a koja sadrži jezgru mrežnog operacijskog sustava koji treba biti pokrenut na klijentu.
- 3) Bootstrap kod na klijentu pokušava, koristeći TFTP mrežni protokol, dohvatiti gore navedenu jezgru operacijskog sustava.
- 4) TFTP server program prosljeđuje klijentu njegovu jezgru mrežnog operacijskog sustava.

- 5) Bootstrap kod na klijentu pokreće dohvaćenu jezgru mrežnog operacijskog sustava te time završava sav svoj posao.
- 6) Pokrenuta jezgra operacijskog sustava ponovo na mrežu šalje DHCP zahtjev u kojem eventualno može tražiti više podataka nego što ih je originalno zatražio i dobio bootstrap kod pri svom inicijalnom kontaktu s DHCP serverom. Ukoliko su jezgri dovoljni podaci koje je originalno od dobio bootstrap kod, onda niti ovaj niti idući korak teoretski nisu potrebni. Ipak, zbog načina rada trenutne implementacije FAI sustava ovaj je korak nužan kako bi taj sustav ispravno funkcionirao.
- 7) DHCP server šalje klijentu tražene podatke. Između ostalih podataka ovdje se nalazi i točna identifikacija stabla direktorija na mreži koju klijentov mrežni operacijski sustav mora iskoristiti kao osnovno (root) direktorijsko stablo.
- 8) Jezgra operacijskog sustava sada, koristeći NFS mrežni protokol, mounta svoje osnovno (root) direktorijsko stablo.
- 9) Dalje se nastavlja standardno podizanje operacijskog sustava te pokretanje skripti FAI sustava na klijentu.

7.1.2.2. PXE bootstrap kod

- 1) Bootstrap kod na klijentu šalje na mrežu DHCP zahtjev.
- 2) DHCP server program prepoznaje klijenta po njegovoj MAC adresi te mu vraća osnovne informacije o njemu (IP adresu, naziv, informacije o osnovnim dostupnim serverima na mreži i sl.) uključivo s nazivom datoteke koju klijent može dohvatiti na mreži, a koja sadrži daljnji kod koji se mora izvršiti na klijentu.
- 3) Bootstrap kod na klijentu pokušava, koristeći TFTP mrežni protokol, dohvatiti gore navedenu datoteku.
- 4) Odgovarajući TFTP server program klijentu prosljeđuje traženu datoteku.
- 5) Bootstrap kod na klijentu pokreće dohvaćenu datoteku te time završava sav svoj posao.

U daljnjem opisu PXE protokola pretpostavljamo da je vraćena datoteka *pxelinux.0* koja se na server instalira u sklopu Debian paketa *syslinux*. Na internetu postoje dostupne detaljne informacije o tome kako se može, korištenjem specijaliziranih biblioteka, napraviti svoj vlastiti kod koji zamjenjuje datoteku *pxelinux.0*, ali taj dio mi nismo dalje proučavali.

- 6) Kod iz datoteke *pxelinux.0* traži TFTP protokolom daljnje konfiguracijske datoteke za PXE. One se nalaze u poddirektoriju *pxelinux.cfg* istog direktorija iz kojeg je originalno nabavljena datoteka *pxelinux.0*. Točno koja se točno konfiguracijska datoteka koristi određuje se slijedećim algoritmom:
 - a) ako postoji koristi se datoteka nazvana po MAC adresi klijentove mrežne kartice zapisanoj u heksadecimalnom brojevnom sustavu
 - b) ukoliko gornja datoteka ne postoji onda datoteka s istim nazivom, ali bez zadnjeg znaka u nazivu
 - c) itd, dok se ne potroše sva slova iz početnog imena konfiguracijske datoteke
 - d) ukoliko do sada nije pronađena odgovarajuća konfiguracijska datoteka onda se traži konfiguracijska datoteka default
- 7) Ovdje je opisano default ponašanje koda u datoteci *pxelinux.0*, a to je ponašanje moguće u nekoj mjeri još modificirati korištenjem dodatnih

opcija proslijeđenih klijentu kroz DHCP mrežni protokol. Za detaljniji opis ovih DHCP opcija vidi dokumentaciju za PXE koja se instalira u sklopu Debian paketa *syslinux*.

- 8) TFTP server program proslijeđuje klijentu traženu konfiguracijsku datoteku.
 - 9) Kod iz datoteke *pxelinux.0* prema podacima u dohvaćenoj konfiguracijskoj datoteci određuje koju jezgru mrežnog operacijskog sustava i od kud treba nabaviti te ju pokušava dohvatiti ponovo koristeći TFTP mrežni protokol.
 - 10) TFTP server program proslijeđuje klijentu traženu jezgru operacijskog sustava.
 - 11) Kod iz datoteke *pxelinux.0* na klijentu pokreće dohvaćenu jezgru mrežnog operacijskog sustava te time završava sav svoj posao.
- Daljnji koraci potpuno su jednaki koracima 6 i više iz pododjeljka 7.1.2.1.

7.2. Konfiguriranje obrade klijentovih lokalnih diskova.

FAI za particioniranje i formatiziranje lokalnih hard diskova na klijentima koristi Perl skriptu *setup_harddisks*. Sama skripta nalazi se na FAI serveru u direktoriju */usr/lib/fai/sbin*, a instalira se u sklopu *fai* Debian paketa.

Ta skripta particionira i formatizira klijentove lokalne hard diskove prema uputama koje se nalaze u konfiguracijskim datotekama u direktoriju *\$FAI/disk_config*. Koriste se konfiguracijske datoteke koje se zovu kao klase kojima pripada klijent. Npr. ukoliko klijent pripada klasi 4GB onda se koristi, ukoliko postoji, konfiguracijska datoteka *\$FAI/disk_config/4GB*.

U daljnjem ćemo dijelu ovog dodatka dati detaljan opis formata u kojem su zadane gore spomenute konfiguracijske datoteke za particioniranje i formatiziranje lokalnih hard diskova na klijentu.

7.2.1. Format konfiguracijskih datoteka

Svi redovi u tim konfiguracijskim datotekama koji započinju znakom *#* smatraju se komentarima te se ignoriraju od strane FAI sustava.

Primjer jedne konfiguracijske datoteke u kojoj se postavlja jedan IDE disk pod nazivom *hda* i jedan SCSI disk pod nazivom *sdb*:

```
> # <Type> <mount point> <Size/mb> [mount options]           [;extra options]
>
> disk_config hda
> primary / 200 defaults,errors=remount-ro ; ext2
> logical /home 100-300
> logical /scratch1 10- defaults,nosuid ; -i 15000 -m 0
> disk_config sdb
> primary /tmp 300-500 rw ;ext2
> primary /backup preserve2 rw
> logical swap 50-100
> logical /scratch2 100-300 rw ; -m 30
> logical - preserve7
> logical /var 100 ; -j
> logical /var/tmp preserve9 ;format
> primary /tmp/mytmp -300
```

Konfiguracija svakog od diskova počinje naredbom *disk_config* iza koje slijedi naziv Linux devicea bez prefiksa */dev/*. Nakon ove naredbe slijedi po jedan red za svaku od particija na tom disku. Tu se za svaku od particija zadaje ukupno pet vrsta podataka:

- Tip particije
- Mjesto mountanja particije
- Veličina particije
- Postavke za mountanje particije
- Dodatne postavke za formatiziranje particije

7.2.1.1. Tip particije

Postoje dva tipa particija: primarna i logička. Primarne particije su bootabilne, ali postoji ograničenje od maksimalno četiri primarne particije na svakom disku. Osnovna (root) particija Linux operacijskog sustava mora biti ovog tipa.

Logičke particije su sve zajedno interno organizirane u jednu primarnu particiju što ima za posljedicu da ukoliko na disku postoji barem jedna logička particija onda osim nje na istom disku može postojati još najviše tri primarne particije.

Nije sigurno da ova podjela na primarne i logičke vrijedi (u cijelosti ili bar djelomično) za SCSI diskove, ali što se formata ovih konfiguracijskih datoteka tiče tip particije mora biti zadan.

Najčešće se na disku postavlja samo jedna primarna particija - root particija, a sve ostale (npr. */tmp*, */usr*, */swap* i */var*) označavaju se kao logičke. Primjer ovakve konfiguracije hard diska je gore prikazana konfiguracija za disk *hda*.

7.2.1.2. Mjesto mountanja particije

Mjesto mountanja particije je puna putanja (uključujući početni */*) za datotečni sustav na toj particiji. Vrijednost "swap" definira particiju kao swap particiju Linux operacijskog sustava. Vrijednost "-" označava da se particija neće mountati na sustavu nego da se može koristiti za druge tipove datotečnih sustava, kao što su VAX, UFS, MINIX i drugi.

Standardne te Linux swap particije bit će zapisane na klijentovom operacijskom sustavu u datoteci */etc/fstab*. Tamo je zapisan statički opis klijentovog datotečnog sustava. Za detaljnije informacije o toj konfiguracijskoj datoteci vidi *man* stranice za *fstab(5)*.

7.2.1.3. Veličina particije

Ovdje je zadana veličina particije u megabyteima. Ova se vrijednost automatski zaokružuje tako da bude ispravna s obzirom na broj cilindara diska.

Postoji nekoliko načina na koje se može zadati ova veličina:

- "200" znači 200MB - ni manje ni više
- "100-300" postavlja minimum od 100MB i maksimum od 300MB.
- "10-" postavlja minimum of 10MB i ne postavlja maksimum.
- "-300" postavlja minimum od 1MB i maksimum od 300MB.
- ("-") nije ispravna vrijednost za veličinu particije.)

Po defaultu na ovoj particiji napraviti će se novi datotečni sustav te će na njoj biti prebrisani svi stari podaci. To se može izbjeći korištenjem opcije "preserve<no>" umjesto veličine particije. Točan opis korištenja te opcije bit će dan malo kasnije.

Ukoliko je kao veličina dvije ili više particija naveden interval onda skripta određuje njihove točne veličine tako da pokušava sačuvati njihovih omjer.

7.2.1.4. Postavke za mountanje particije

To su postavke koje će biti zapisane u */etc/fstab* konfiguracijsku datoteku na novoinstaliranom klijentovom operacijskom sustavu. Prazna vrijednost postavlja ove postavke na "defaults". Za detaljnije informacije vidi *man* stranice *mount(8)* i *fstab(5)*.

7.2.1.5. Dodatne postavke za formatiziranje particije

Zadnje polje predviđeno je za eventualno zadavanje dodatnih postavki za formatiziranje dane particije klijentovog lokalnog diska. Ukoliko se zadaje, onda mora od prethodnog dijela konfiguracije za tu particiju biti odvojeno znakom ;.

Neke od mogućih opcija su slijedeće:

- **-i <bytes>** - broj byteova po inode-u (samo za *ext2* i *ext3* datotečne sustave)
- **-m<blocks>** - postotak blokova rezerviran za superusera (samo za *ext2* i *ext3* datotečne sustave)
- **-j** - stvaranje datotečnog sustava s *ext3* dnevnikom.
- **-c** - provjeravanje cijele particije za neispravne blokove.
- **ext2** - označavanje particije kao *ext2* u */etc/fstab*.
- **ext3** - označavanje particije kao *ext3* u */etc/fstab*.
- **format** - formatiziranje ove particije, čak i ako je navedena opcija *preserve* (vidi dolje).
- **reiser** – stvaranje *Reiser* umjesto *ext2* datotečnog sustava.

Redoslijed samih opcija nije bitan, a za više informacija o njima pogledajte *man* stranice za program *mke2fs(8)* koji interno koristi ova skripta FAI sustava za formatiziranje particija na disku.

Iz gore navedenog slijedi da imamo slijedeću interakciju među postavkama **-j**, **ext2** i **ext3**:

- **<bez postavke>** - *ext2* datotečni sustav označen kao *auto* u */etc/fstab*
- **-j** - *ext3* datotečni sustav označen kao *auto* u */etc/fstab*
- **ext2** - *ext2* datotečni sustav označen kao *ext2* u */etc/fstab*
- **-j ext2** - *ext3* datotečni sustav označen kao *ext2* u */etc/fstab*
- **-j ext3** - *ext3* datotečni sustav označen kao *ext3* u */etc/fstab*
- **ext3** - *ext2* datotečni sustav označen kao *ext3* u */etc/fstab*

Korištenje opcije *auto* u */etc/fstab* konfiguracijskoj datoteci omogućuje operacijskim sustavima koji ne podržavaju *ext3* datotečne sustave da se pokušaju nositi s njima kako najbolje znaju, npr. tako da ih tretiraju kao *ext2* datotečne sustave.

7.2.1.6. Zadržavanje particija

Moguće je sačuvati veličinu particije te ako ih ima i podatke spremljene na njoj. Da bi smo sačuvali veličinu particije broj particije mora ostati nepromijenjen, a njena veličina mora biti navedena kao "*preserve<no>*". Broj

<no> je broj devicea te particije (za točno odrediti koji device pripada toj particiji vidi izlaz naredbe *df* na lokalnom klijentovom operacijskom sustavu). Primarne particije imaju brojeve od jedan do četiri dok brojevi logičkih particija počinju od 5.

Evo primjera u kojem su uz ostale podatke o particijama prikazani brojevi njihovih devicea za disk *sdb* (vidi primjer s početka):

primary	/tmp	300-500	#	1
primary	/backup	preserve2	#	2
logical	swap	50-100	#	(3) 5
logical	/scratch2	100-300	#	(3) 6
logical	-	preserve7	#	(3) 7
logical	/var	100	#	(3) 8
logical	/var/tmp	preserve9	#	(3) 9
primary	/tmp/mytmp	-300	#	4

Prve dvije particije su primarne (tipa *primary*) pa dobivaju brojeve 1 i 2. Logičke particije dobivaju brojeve od 5 na gore, a u slučaju diska *sdb* iz gornjeg primjera postoji 5 logičkih particija pa zadnja dobiva broj 9, a sve one zajedno interno su grupirane u jednu primarnu particiju s brojem 3. Dakle ukoliko želimo sačuvati logičku particiju s brojem 7 moramo ubaciti pred nju točno dvije logičke particije.

7.3. Neke korisne WWW stranice

Linuxu i njegove distribucije:

<http://www.linux.org>

<http://www.linux.com>

<http://www.debian.org>

<http://www.freshmeat.com>

<http://www.kernelhq.com>

<http://www.kernel.org>

<http://www.linuxapps.com>

<http://www.tucows.com>

<http://www.linuxworld.com>

Podizanje operacijskih sustava preko mreže:

<http://netboot.sourceforge.com>

<http://etherboot.sourceforge.com>

<http://www.ltsp.org/>

FAI sustav:

<http://www.informatik.uni-koeln.de/fai>

8. Literatura

Knjige:

'*Linux administration handbook*', Evi Nemeth, Garth Snyder, Trent R. Hein; Upper Saddle River, NJ 07458, Prentice Hall PTR. 2002.

'*Running Linux, 3rd Edition*', Welsh, Matt, Matthias Kalle Dalheimer and Lar Kaufman; Sebastopol, CA: O'Reilly and Associates. 1999.

'*Debian GNU/Linux Bible*', Steve Hunger; Hungry Minds, Inc, 2001.

Članci i knjige na internetu:

'*FAI Guide (Fully Automatic Installation)*', Thomas Lange, <http://www.informatik.uni-koeln.de/fai/fai-guide.pdf>

'*PXELINUX*', H. Peter Anvin

'*Bootstrapping an Infrastructure*', Infrastructures.org Project, <http://www.infrastructures.org/papers/bootstrap/bootstrap.html>

'*Network Boot and Exotic Root HOWTO*', Brieuc Jeunhomme, Logilab S.A.

'*The Debian Maintenance HOWTO*', Martin Schulze

'*The Debian GNU/Linux FAQ*', <http://www.debian.org/doc/FAQ>

'*Filesystem Hierarchy Standard – Version 2.2 final*', Filesystem Hierarchy Standard Group