Predavanje 1 - Uvod

Mrežna i sistemska administracija je inženjerska disciplina koja se podjednako bavi tehnologijom računarskih sistema kao i korisnicima te tehnologije.

Tiču se sastavljanjem mreže, radom te mreže, te održavanjem mreže uprkos aktivnostima koje korisnici izvršavaju, a koje dovode do kvara na mreži.

Sistem administrator radi za korisnike, tako da oni mogu koristiti mrežu za obavljanje svojih poslova, ali isto tako pruža usluge za zajednicu korisnika. Danas, pod pojmom zajednica se podrazumijeva globalna zajednica mašina i organizacija, upravo zahvaljujući Internetu.

Administracija računarskih mreža nije (samo) administrativni posao (uprkos nazivu) već je to zahtjevan inženjerski posao.

Zadaci

Zadaci: hardver, softver, podrška korisnicima, dijagnostika, popravke i prevencija. System administrator mora da posjeduje tehničke, administrativne i društveno psihološke vještine.

Glavni zadaci i network i system administratora su hardverska konfiguracija, konfiguracija softverskih sistema, a oba ova zadatka su za korisnike. Hardver zahtjeva napajanje, odgovarajuću temperaturu i vlažnost kako bi mogao da radi. Softver zahtjeva hardver, za svoj rad, tj. hardver ograničava softver. Softver nema fizičkih interakcija sa okolinom.

Računarska mreža i usluge koje pruža imaju svrhu (poslovna, obrazovna, ...) koja rukovodi postupke i odluke administratora.

Administracija računarskih mreža zahtjeva strpljenje, razumijevanje, znanje i iskustvo.

Etička pitanja

Iz knjige:

Postoji mnogo etičkih pitanja u administraciji računarskih mreža. Iako se neke odluke mogu donijeti objektivno (npr. povećanje produktivnosti ili smanjenje troškova), ipak treba da postoji politika (policy) koja se tiče upotrebe i menadžmenta kompjutera i njihovih korisnika.

Neke odluke se moraju donijeti kako bi se zaštitila prava individua. Sistem administrator mora da uzme u obzir dosta odgovornosti i dosta ograničenja. Prva odgovornost je odgovornost prema zajednici, a onda prema korisnicima.

Mnogo odgovornosti, mnogo ograničenja, pristup (privatnim, poslovnim) informacijama, odluka o prioritetima(donosi vlasnik infrastrukture), politika.

Izazovi

System administration se ne bavi samo instaliranjem OS-a. Tiče se planiranja i dizajniranja učinkovite zajednice računara, tako da korisnici mogu da obavljaju svoje poslove.

Izazovi administracije računarskih mreža:

- Dizajn logične i efikasne mreže
- Priključivanje velikog broja računara sa mogućnošću budućeg lakog ažuriranja i proširenja
- Utvrđivanje potrebnih usluga
- Planiranje i realizacija odgovarajuće sigurnosti
- Stvaranje ugodnog okruženja za korisnike
- Razvoj metoda otklanjanja grešaka i problema
- Upravljanje znanjem, potrebnim (i drugim)

Praksa

Postoje 3 razloga zašto se neka praksa zadrži:

1. Neko je to uradio u prošlosti, ideja je kopirana bez razmišljanja i od tad niko ne razmišlja o tome već svi je primijenjuju.
2. Stručnjaci su duboko analizirali i ponudili zaista najbolje rješenje.
3. Moralo se izabrati rješenje, koje je sada prihvaćeno.

Bugovi

Operativni sistemi i programi su puni bugova koji nisu planirani ili dizajnirani. Oni mogu nastati iz više razloga:

1. Loša kontrola softvera ili procedura
2. Problemi u OS ili njegovim subsistemima
3. Problemi nastali uslijed nekompatibilnih softvera
4. Neobjašnjivi fenomeni, virusi i ostali napadi.

Administrator mora biti u mogućnosti da radi uprkos ovim problemima jer se ne mogu svi problemi riješiti na izvoru istih (nemamo izvorni kod ili nemamo pravo na izvorni kod).

Principi

Sistemska administracija kreće od *politike* – odluke o tome šta želimo i kako treba da bude s obzirom na to što možemo priuštiti Svaka računarska mreža ima svoju politiku, koja opisuje namjene i svrhu mreže *Predvidljivost* – Administrator treba da teži predvidljivom sistemu koji je osnova pouzdanosti, te povjerenja i stoga sigurnosti Kako mreže kompjutera i korisnika rastu, njihove interakcije postaju veoma kompleksne, samim tim i nedeterminističke (nepredvidive u okvirima nekog “managable” broja varijabli). Zbog toga je treći princip skalabilnost. *Skalabilnost* - Skalabilni sistemi mogu rasti u skladu sa politikom, oni se ponašaju predvidljivo čak i kad se povećavaju.

- Jednostavnost rješenja
- Jasnoća rješenja
- Opštost rješenja
- Automatizacija zadataka

● Komunikacija sa ljudima
● Prvo osnovne stvari

Znanje

Poznavanje savremenih tehnologija zastarjeva jednako brzo kao i tehnologije. Stara znanja treba odbacivati i sticati nova. Ovo je beskonačna (i naporna) petlja. Pravo (trajno) znanje je zasnovano na razumijevanju (Knowledge begins with understanding).

Pitanje 1: Zbog čega je bitno definisati šta se smatra hitnim?
Hitnost pomaže da se između više stvari koje je neophodno uraditi, prvo rade one koje su definisane kao hitne. Time se olakšava administratoru organizacija.

Pitanje 2: Zašto je bitno napraviti skalabilan sistem?
Skalabilan sistem raste na način da taj rast ne utiče negativno na njegovo funkcionisanje, odnosno ne zahtjeva velike zahvate na sistemu.
Potrebno je da on raste na predvidiv način.
Skalabilan sistem se ponaša predvidivo pri njegovom rastu tj. Kad sistem raste, ako je skalabilan, on će se i dalje ponašati predvidivo.

Pitanje 3: Šta je sistemska politika?
Odluka o tome šta želimo i kako ćemo se ponašati u zavisnosti od toga šta možemo sebi priuštiti.
Potrebno je da admin sistema zna sa čime raspolaže i koliki njegov sistem treba da bude, šta treba da podržava, te na osnovu toga da napravi policy.
Treba znati čemu služi i čime raspolaže.
Primjer: youtube, superbowl

Pitanje 4: Šta je predvidiv sistem?
Kako će se sistem ponašati u različitim situacijama.
To nam pomaže u skraćivanju vremena pronalaženja problema i rješavanja problema.
Znam šta mogu očekivati da će se desiti.
Jedna od osnova napada je da se sistem dovede u situaciju koja nije predviđena.

Pitanje 5: Da li je administracija računarskih mreža administrativni ili inžinjerski posao?
Oboje.
Administrativni jer administrator mora da posjeduje organizacione, administrativne, te društveno-psihološke osobine.
Inžinjerski je jer su potrebno konfigurisati hardver, konfigurisati softverski sistem, baviti se dijagnostikom, popravkama i održavanjem.

Pitanje 6: Fizičko okruženje za računarske mreže?
Hardveru je potrebno fizičko okruženje jer ako nema odgovarajuću temperaturu, vlažnost, napajanje itd. neće raditi ispravno (neće biti predvidivo).
Rad HW utiče na rad SW.

Pitanje 7: Zašto je administratoru teško da ukloni sve bugove u SW?
Potrebno je zaobići bug koji ne možemo ukloniti.
Ne može otkloniti sve bugove, jer nemamo izvorni kod, a čak i ako imamo, možda nemamo pravo na to.
Moramo našu mrežu prilagoditi takvom ponašanju tih softvera.

Pitanje 8: Etička pitanje?
Prvo pitanje je privatnost korisnika (zato što administrator ima pristup svim podacima), zatim povjerljivost podataka.

Pitanje 9: Zašto je bitno utvrditi(?) usluge tj. Zašto administrator treba da zna koje usluge treba da pruža korisniku?
Na osnovu usluga možemo definisati prioritete usluga.
Usluge koje moramo pružati definišu kakvi nam resursi trebaju (usluge definišu šta nam treba, koji hw, koji sw).

Pitanje 10: Ko definiše prioritete?
Vlasnici sistema - onaj ko od toga živi. Administrator ne bi trebao biti taj ko definiše prioritete.

Pitanje 11: Zašto je važno znati šta ne znamo?
Da znamo da trebamo naučiti

Pitanje 12: Opštost rješenja?
Šta se desi kad riješimo konkretan problem? Ako riješimo konkretan problem, kasnije nećemo znati riješiti problem iz iste klase problema
Zašto je bitno dobro utvrditi koje usluge računarska mreža treba da pruža korisnicima?
Da bi odredili kakvi resursi trebaju unutar mreže (hardverski i softverski), potrebno je da se kvalitetno odredi kakve usluge ta mreža treba da pruža korisnicima, da bi tti resursi mogli da iznesu ono što treba da pružaju korisnicima. Ovo spada u samo planiranje mreže.
Koja su etička pitanja koja se postavljaju pred administratore računarske mreže?
Generalno, administratori mreža, u svakom trenutku imaju pristup svim podacima na mreži. Etička pitanja koja se postavljaju pred njega leže u tome, šta on sa tim pristupom tim podacima treba ili ne treba da radi. Administratori mreža treba da se ponašaju u skladu sa nekim etičkim kodeksom ponašanja, u koji spada ne zadiranje u privatnost korisnika, iskorištavanje njihovih podataka u loše ili kriminalne svrhe i slično.

Predavanje 2 - Komponente računarske mreže

Od čega se sastoji human-computer system:

1. Ljudi(korisnici) - koriste infrastrukturu i uzrokuju najviše problema. Korisnici utiču na ponašanje mreže i upotreba resursa zavisi od ljudskih navika.
2. Računari - na njima se izvršavaju programi
3. Mrežni uređaji (routeri, switchevi, kablovi)

Elektronski uređaji su osjetljivi.

Zašto treba gasiti računar? OS ima određene funkcije na shut down-u. Svi procesi koji trebaju da se završe, završe se gašenjem računara.

Zašto treba safely remove USB? Kad prebacujemo podatke na USB, može se desiti da je OS zauzet i da će odraditi prebacivanje malo kasnije. Ako mi izvučemo USB, može se desiti da podaci ne budu zapisani.

Poželjno je da se, prilikom upravljanja hardverom, pročitaju instrukcije za korištenje, jer one su napisane s razlogom. Na taj način možemo spriječiti neadekvatno upravljanje hardverom i spriječiti kvarove. Hardver treba zaštititi od okoline na koju mi ne možemo da utičemo. HW ne voli nagle promjene napona (visoki naponi, npr. munja). Ako hoćemo pružiti ozbiljnu podršku korisnicima, moramo se brinuti o HW.

Klima osigurava adekvatne uslove po pitanje temperature i vlažnosti prostorije. Ukoliko je prostorija pretopla, može doći do pregrijavanja i prestanka rada hardvera. Ukoliko prostorija nije dovoljno vlažna, povećava se količina statičkog elektriciteta. S druge strane, ukoliko je prostorija prevlažna, može doći do hrđanja.

Hard diskovi:

-2 najrašireniji tipa diskova su ATA (bivši IDE) i SCSI. ATA diskovi su jeftiniji od SCSI. Kod ATA diskova je pristup sekvencijalni. S druge strane, SCSI diskovi su učinkovitiji u multiple-access i stoga su bolji u multi-tasking sistemima, gdje je bitan random access.

-Ne možemo staviti novi disk u stari računar jer se ne mogu miješati različiti interfejsi. Neophodno je kupiti odgovarajuće rezervne dijelove.

-Stari diskovi su IDE, odnosno paralelni ATA diskovi. Savremeni diskovi su serijski.

Kod serijskih diskova se prenosi bit po bit, a kod paralelnih se prenosi više bita istovremeno.

-I kod ATA i kod SCSI su pobijedili serijski.

-U korisničkim računarima je SATA (serial ATA). SCSI je nekad preovladavao u serverskim, a danas su u serverskim SAS ili fibre channel.

-Još uvijek postoji veliki broj HDD, koji su sad jeftini i rade solidno, za neku masovnu upotrebu. Danas je puno bolje se orjentisati ka SSD, oni nemaju mehaničkih dijelova i pouzdaniji su. Nisu za masovnu pohranu podataka, na njih je najbolje staviti OS i aplikacije, jer je to ono što sporije radi. Prije je memorija bila ograničavajući faktor u smislu da je predstavljala usko grlo sistema, dok su danas to diskovi.

IDE - Integrated Drive Electronics; SATA - Serial Advanced Technology Attachment

Oba interfejsa predstavljaju način konektovanja uređaja za spremanje podataka (HDD) na kompjutersku sistemsku sabirnicu.

Razlika između IDE i SATA interfejsa:

IDE je paralelni ATA. SATA je serijski ATA.

IDE je stariji standard od SATA-e. (IDE stvoren 1986, SATA 2003)

IDE driveri su sporiji od SATA drivera.

IDE interfejs ne podržava hot-plugging dok SATA interfejs podržava.

IDE interfejs podržava prenos podataka do brzine od 133MB/s. SATA pruža brzinu prenosa podataka do 6Gb/s.

Memorija:

Treba gledati kapacitet i brzine. Brzine utiču na mogućnost kombinovanja različitih memorijskih modula. Kapacitet - zato što moramo voditi računa da li može stati u sistem.

Slotovi za proširenja:

PCI express je interfejs koji se koristi za plugovanje modernih extension kartica u moderne kompjutere ili matične ploče. Zamijenio je AGP i PCI. AGP je interfejs koji se može koristiti samo za grafičke kartice dok je PCI korišten za sve ostale vrste kartica. PCI Express se koristi za svaki vid kartica (grafičke, zvučne, mrežne, itd.) te predstavlja dominantan interfejs na tržištu.

Operativni sistemi

-Operativni sistem ima nekoliko ključnih elemenata, a to su: dio zadužen za upravljanje hardverom (diskovima, tastaturom, ekranom i sl), filesystem(koji omogućava organizovanje file-ova) i user interface.

-Da bi se na hardveru izvršavao bilo kakav softver, neophodno je da imamo operativni sistem.

-Aplikacije koje koriste korisnici ne mogu direktno pristupiti uređajima, zato služi operativni sistem. OS treba da omogući interfejs aplikacijama da one mogu pristupiti bilo kakvom uređaju (disku, ekranu itd).

-OS se mogu podijeliti na operativne sisteme koji mogu da rade samo jednu stvar istovremeno (singletasking - DOS) i više stvari istovremeno (multitasking - Unix, NT itd).

Također, mogu se podijeliti prema broju korisnika na jednokorisničke (single user) i višekorisničke (multi user). Svi savremeni OS su višekorisnički. Unix su višekorisnički od početka, a Windows od NT verzije

Unixoidni OS

Postoje rane varijante Unix sistema, a većina vodi porijeklo od BSD(OpenBSD, FreeBSD,...), a drugi su od Systema V.

Zbog raznolikosti tih sistema, usvojen je POSIX standard koji mogućava da svi Unix sistemi koji poštuju ovaj standard rade na isti način.

Iz Unixa je nastao Linux koji je u potpunosti otvoren i besplatan. Linux je dominantan serverski operativni sistem, 67.8% od svih registrovanih server mašina na svijetu koristi Linux. Linux definitivno nije dominantan na području desktop računara, 1.4% desktop mašina na svijetu je registrovano da koristi Linux. Fun fact: najrašireniji operativni sistem svijeta je Android kada se uzmu u obzir mobilni uređaji.

MAC OS X, Android iOS su također Unixoidni OS.

Microsoft

Drži ogroman dio tržišta i zbog toga se ne može zanemariti. Neke stvari su Unixoidni sistemi puno bolje napravili, ali danas na Win10 imamo bash, tj. komandno okruženje Unix-a. Windows je puno bolji u alatima.

Jedna od stvari u kojoj je Unix bolji od Windowsa je što se može samo proces resetovati, a kod Windowsa restartovati sistem(zato je kod servera puno češće korišten Unix, a ne Windows).

Dugo vremena je problem kod Windowsa bilo to što su računari bili jednokorisnički, jednokorisnički sistem, izrazito jednostavno. Imali su problem sa sigurnosti.

Višekorisnički sistemi i sigurnost

Fundamentalni preduslov za sigurnost je mogućnost zabrane pristupa određenim sistemskim resursima. Obični korisnik ne bi trebao da ima pristup file-ovima operativnog sistema, već samo svojim file-ovima (i programima), jer na taj način on me može da ugrozi sigurnost cijelog sistema. Administratori mreža moraju da imaju pristup cijelog sistema, s ciljem nadgledanja istog, pravljenja backup-a i održavanja. Radi ovoga, mora da postoji privilegovani account (za privilegovanog korisnika). Na Windowsu se privilegovani korisnik zove Administrator, a na Unixoidnim sistemima se zove root (ne mora se zvati administrator, ni root, može imati drugo ime). To je korisnik koji ima sve privilegije. Administrator ni root ne bi trebali da se koriste za normal work, zato što privilegovani korisnik ima velike mogućnosti.

Druga bitna stvar je pitanje sigurnosti Windowsa i Linuxa. Sigurnost najviše izlazi iz ispravne konfiguracije.

Datotečni sistem

Podaci koje mi zapisujemo na disk, zapisuju se kao nizovi 0 i 1. Da bi oni bili organizovani u neke datoteke, tj. da bi bile posložene u neki smisleni niz bita koji predstavlja podatke, neophodno je da postoji komad softvera koji će omogućiti organizovanje niza bita u datoteke i to je datotečni sistem. Nas zanima u čemu se razlikuju različiti datotečni sistemi.

Funkcija datotečnog sistema, pored gore navedene organizacije bita u datoteke (zapise) je upravljanje pravima pristupa (ako imamo više korisnika, određuje se koji korisnik može da pristupi kojoj datoteci).

Unixovi datotečni sistemi:

-Unix ima datotečni sistem organizovan hijerarhijski. Direktoriji i subdirektoriji formiraju stablo.

S ciljem zabrane pristupa sistemskim file-ovima, Unix ima informacije o tome ko kreira file-ove i ko ima pravo pristupa. Svaki user ima svoj username, kao i user id(uid). Ako je korisnik A kreirao neki file, owner tog file-a je korisnik A i on može odrediti ko ima pravo da čita(r), piše(w) ili izvršava(x) file.

-Na Unixoidnim sistemima postoje grupe korisnika. Ideja grupe je da omogući određenim korisnicima da čitaju i rade na nekim file-ovima, bez da drugi korisnici to mogu da vide. Svaki korisnik je član barem jedne grupe koja se naziva login group. Svaka grupa ima svoje ime i group id (gid). User id (uid) i group id (gid) svakog korisnika je zapisan u file /etc/passwd. Grupu može kreirati superuser, tako što edituje file /etc/group.

-ext4 se najviše koristi danas

-SWAP je dio datotečnog sistema koji je vezan za virtuelnu memoriju (Swap file, hibernacija), to je posebna particija kod Linuxa

-Omogućava simboličke i stalne veze (linkovi)

-Provodi kontrolu pristupa

-Ima nešto jednostavniju i ograničeniju kontrolu pristupa nego Windows

Windows datotečni sistemi:

-Stariji datotečni sistem je bio FAT i on je veoma nesiguani, u smislu da nije postojao nikakav mehanizam da zabrani pristup file-ovima. To ima za posljedicu da npr. Na SD karticama(organizovane kao FAT), sve aplikacije imaju pristup svemu što pišemo na tu particiju. Razlog zašto je SD kartica formatirana kao FAT: FAT je najšire podržan od različitih uređaja, gdje god stavimo SD karticu, podaci se mogu pročitati.

-Današnji datotečni sistem je NTFS i krenuo je sa NT verzijom, a cilj je da se riješi problem pristupa. NTFS, je kao i Unixov datotečni sistem organizovan hijerarhijski, sa file-ovima i direktorijima.
-Kod Windowsa, disk se dijeli na nezavisne particije i svaka je prikazana kao nezavisna. Obilježavaju se slovima iz alfabeta i kreće se od C.
-Ako imamo korisničke dokumente, zgodno je sve držati u jednom folderu. Kad "selimo" s jednog računara na drugi(ili sa jednog OS na drugi), treba to sve prebaciti, pa je lakše kad je sve u jednom folderu.
-Razlika između Windows-a i Unix: kod Windowsa, ekstenzije datoteka određuju tip datoteka. OS na osnovu ekstenzije ima preslikavanje koje je zapisano u registrima sa kojom će aplikacijom otvoriti koju datoteku. Kod Linuxa datoteka ima tip.
-Kod prava pristupa, kod Windowsa postoji Access Control List. Kod Linuxa postoji za svaku datoteku, pravo pristupa ima njen vlasnik, grupa vlasnik(?) i svi ostali. Kod Windowsa za svaku datoteku, postoji spisak svih korisnika na računaru i grupa, te kakva prava svaka od njih ima.
Mrežni datotečni sistemi:
-SMB(CIFS) je protokol koji omogućava razmjenu datoteka. Kad se napravi shared folder koji se dijeli sa drugim korisnicima na mreži, koristi se SMB. Podržan je od strane Windowsa i od Linuxa. Iz tog razloga je moguće na mreži napraviti folder koristeći ovaj protokol, kojem će moći pristupiti i korisnici Windowsa i korisnici Linuxa.
-Mrežni datotečni sistemi ustvari omogućavaju da različiti OS razmjenjuju datoteke koristeći SMB protokol.
-Ima slične funkcionalnosti kao FTP, ali je integrisan u OS.

Procesi

-Svaki program koji pokrenemo, pokrenemo datoteku koja generiše jedan ili više procesa. Sve što radimo je proces i svaki proces se izvršava pod prijavom korisnika koji ga je pokrenuo. Procesi se izvršavaju u pozadini (background) i foreground (procesi s kojim imamo interakciju).

Procesi na Unixoidnim sistemima

-Svaki proces ima svoj ID (PID). Da izlistamo sve procese, možemo koristiti komandu *ps* .
-Kada pokrenemo proces, novi proces postaje dijete originalnog procesa. Dakle, procesi formiraju hijerarhiju. Više djece može da ima istog roditelja. Kod Unixoidnih sistema, svi procesi su djeca inicijalnog procesa *init* , koji ima PID=1.
-Ako ubijemo roditelja, onda su sva njegova djeca ubijena, osim ako dijete nije uspjelo da se "otkači" od roditelja. Ako dijete umre, to ne utiče na roditelja. U nekim situacijama, dijete kad umre postaje zombi proces. To znači da to dijete-proces ima roditelja koji čeka da taj proces završi. Ako roditelj još uvijek nije obaviješten da je dijete ubijeno, onda se to dijete ne uklanja iz "kernel's process table". Kada se roditelj obavijesti da je dijete ubijeno, djetetov proces-entry se uklanja.
-Varijable okruženja - jedna od njih je putanja, tj. Path koja služi da prikaže redoslijed foldera koji će se pretraživati u potrazi za nazivom neke datoteke

Računarska mreža

-Postoje različiti načini povezivanja računara. Jedni od njih su gridovi. Zašto grid nije računarska mreža?
-RM je međusobno povezana skupina autonomnih računara. Bitno je ovo "autonomni računar", računar radi sam za sebe. Svaki od računara nezavisna jedinka i ne trebaju mu ostali računari da bi funkcionisali. Povezani su jer jedni drugim nude neke usluge.
Ako radimo na nekoj podršci postojanja RM, ljudi smatraju da je to stvar koja mora da radi.

OSI Model

-Fizički sloj: neko je trebao definisati kako izgleda mrežni adapter, kako su definisani pinovi, postoji standard koji je definisao kako izgleda kabal, kakve su parice, kako izgledaju električni signali na kablu, naponski nivoi?
-Sloj veze podataka - ovaj sloj osigurava da nešto što se pošalje od jednog ka drugom kraju kabla, stigne do odredišta. Ovo se naziva handshaking.
Kad postoji kabal kojim su povezani računari, ovaj sloj se brine da on zna prepoznati 0 i 1 koje putuju po tom kablu, i da zna prepoznati adresu na fizičkom sloju, kad podaci putuju od jednog do drugog računara, te da zna ako je došlo do greške da to zaključi. Dakle, ovaj sloj obezbjeđuje povezivanje lokalnih računara.
-Mrežni sloj omogućava uvezivanje mreže.
-Transportni sloj služi da dva računara razmjenjuju informacije (aplikacije komuniciraju).
-Sesijski sloj i prezentacijski sloj ne postoje u TCP/IP, nego su integrisani u aplikacijski sloj. Sesijski sloj omogućava SSL, uspostavljanje sesije. Prezentacijski - transformacija iz jednog načina u drugi način kodiranja.
-Aplikativni sloj
Fizički sloj:
-Današnji kablovi su uglavnom bakarni (reda 100ak metara). Postoje:
● ravni koji se ne koriste

● upredena parica nekad su se koristili, to je npr. telefonski kabal sa 4 žice, ali se ne koristi za mreže
● koaksijalni - sa povećanjem broja kablovskog pristupa se koristi za prenos mrežnih signala. Danas se koristi za povezivanje domaćinstva sa providerom, ali u principu ne koristi se za prenos mrežnih podataka. Danas ili bežični ili ethernet.
Fiber optički kablovi, u odnosu na bakarne omogućavaju puno veću brzinu i puno veće udaljenosti.
Danas je u domaćinstvima dominantan bežični prenos, ali je malo nepouzdanije. Bežičnom prenosu smetaju druge mreže. Na jednoj frekvenciji jedna informacija.
Sloj veze podataka:
-Ethernet je dominantna tehnologija.
-Token kao tehnologija je izrazito efikasna. Bolje je koristila medij nego ethernet, ali je ethernet bio dovoljno dobar i imao je nižu cijenu. Uvijek je u historiji pobijedila tehnologija koja je bila dovoljno dobra i koja je imala nižu cijenu. Ne treba "ganjati" ono što je savršeno, ako nije povoljno.
-Uz Ethernet i Wireless, sa 1Gb/s (brzina opada sa udaljavanjem)
-I ethernet i wireless koriste dijeljeni mediji. Treba da postoji način da se podijeli vrijeme između kad svi koriste dijeljeni medij. MAC (Media Access Control) su protokoli koji upravljaju pristupom mediju.
Povezivanje:
-Repeater, Hub i Bridge su stari uređaji. Danas se koristi switch.
-Repeater - pojačavanje signala, nedostatak je sto je duplo gtsporiji.
- **Hub** je mrežni uređaj koji spaja više mašina u jednu mrežu. Veoma je jednostavan ali troši mnogo bandwitha. Hub radi na principu da kada primi paket za slanje od jednog uređaja u mreži, taj isti paket replicira u onoliko kopija koliko je uređaja spojeno na hub (ne računajući mašinu koja šalje poruku). Kreirane kopije dolaze do svih uređaja u mreži gdje paket prihvata onaj uređaj kome je poruka namijenjena dok ostali paket ignorišu. Hub je veoma jeftin i jednostavan način kreiranja mreže ali mu je glavna mana što kreira nepotreban promet na mreži. Hub je **sloj 1 uređaj** što znači da "ne posjeduje znanje" o adresama. Ono što hub radi jeste da replicira bite koji dolaze do njega te ih šalje na ostale uređaje u mreži. Hub ne može primati i slati podatke u isto vrijeme.
-Ne mora biti preslikavanje funkcionalnosti i uređaja 1 na 1. Obično jedan uređaj ima više funkcionalnosti.
- **Modem** (modulator/demodulator) služi za konverziju signala analognog u digitalni.
- **Bridge** je mrežni uređaj sloja 2, što znači da razumije da je u stanju učiti MAC adrese uređaja koji su spojeni na njega. Bridge se koristi da segmentira lokalnu mrežu (LAN) u više dijelova (uveden da smanji promet koji je postao prezasićen korištenjem Huba u mreži). Bridge može slati i primati podatke u isto vrijeme.
- **Switch** je mrežni uređaj koji kombinuje funkcionalnosti huba i bridgea. Povezuje uređaje u mreži te je u stanju da nauči koji port je konektovan na koju mašinu (eng. host). Switch posjeduje MAC-PORT tabelu. U datoj tabeli su zapisane informacije o tome koji port je spojen na koji uređaj (MAC adresa spojenog uređaja). Switch je mrežni uređaj drugog sloja. Switch, u usporedbi sa bridgeom i hubom, znatno smanjuje bandwith mreže te implementira veći stepen sigurnosti budući da paketi stižu samo do odredišne mašine a ne svih mašina u mreži.
- **Router** je mrežni uređaj sloja 3, što znači da ruter pored MAC adresa koristi i **IP adrese.** Ruter je uređaj koji spaja dvije mreže. Ruter predstavlja gateway za sav promet koji izlazi iz lokalne mreže ka internetu.
Osim ovoga, router sprječava širenje poruka iz mreže, koje druge mreže ne bi trebale da vide. Također, router služi da filtrira neželjeni saobraćaj iz sigurnosnih razloga.
Podjela mreža:
Mreže se dijele na lokalne (pod kontrolom i vlasništvom jedne organizacije koja provodi politiku, upravlja ruterima i sl) i globalne.
Virtuelni LAN - omogućavaju da ako imamo switch, svi računari koji su "utaknuti" na switch su dio istog LAN-a, možemo napraviti tako da npr. 3 računara budu 1 LAN, a ostali su drugi LAN. Omogućava stvaranje privida da postoje 2 fizička switcha, a zapravo postoji 1.
Protokoli:
Da bismo mogli razmjenjivati bilo kakve informacije neophodno je da postoji dogovor oko toga kako se te informacije trebaju razmjenjivati. Ako ne postoji protokol, ne postoji način komunikacije.
Skup protokola koji danas dominira je TCP/IP.
IPv4
-4 bajta, 32 bita podataka
-U principu, adrese su i dalje jedinstvene, ali ih nema dovoljno za sve uređaje.
-Hijerarhijski organizovane - podijeljeno je po providerima, svjetska organizacija koja upravlja IP adresama i domenskim imenima dodijeli regionalni registar za Evropu, koji ima IP adrese za Evropu i onda taj registar dodjeljuje providerima. Provideri dobiju svoj skup

adresa.
-Namjena: IP adresa služi da kad paket krene od jednog računara, dođe do drugog računara. Dođe do providera, pa od providera do nas.
-Javne adrese su jedinstvene, a skupovi privatnih adresa koji se mogu koristiti jedino unutar zatvorenog okruženja (paketi čija su odredišta te adrese neće biti proslijeđene).
-Naš računar je podešen da automatski dobije adresu od DHCP, a ako nema odgovora, tad postoji skup adresa koje automatski dobijemo. Ako imamo 5 računara koji su povezani na isti switch i na svima je podešeno da im DHCP dodjeljuje IP adresu, oni će pokušati na taj način dobiti. Ako ne dobiju, svi će dobiti različitu adresu iz ovog skupa. (?)
-Jedan računar ima onoliko adresa koliko ima interfejsa. Ako ima 2 mrežne kartice(interfejs), on ima 2 IP adrese.
-Teoretski, može biti više IP adresa na jednom interfejsu. IP adresa je adresa interfejsa (ne računara!)
-Ruteri povezuju različite mreže i imaju više interfejsa, zato što su povezani na više mreža. Za svaku mrežu na koju su povezani, oni imaju jedan interfejs u toj mreži.

Podmreže:

-Cijeli internet je povezan u podmreže.
-Komunikacija između čvorova u podmreži se odvija bez posredstva rutera. Da bi se saznalo da li su dva računara u podmreži, koriste se podmrežne maske. Podmrežna maska služi računaru da može automatski da zna koji su računari u njegovoj podmreži.
-IP adresa - niz 1, pa niz 0. Računari koji su u istoj podmreži na tim pozicijama gdje su 1, njima odgovaraju iste vrijednosti.
-255.255.255.0 (24 jedinica), svi računari kod kojih su prva tri okteta ista su u istoj podmreži.
-Logičko podešavanje treba da odgovara fizičkom podešavanju, kako bi računari znali da li su neki računar u njegovoj mreži.
-Koliko podmreža - izvadimo ruter, “ostrva” koja su ostala povezana su podmreže (?)
-Primjer na 35. slide-u ima 6 podmreža
-Način adresiranja: prva adresa je adresa mreže koja ne bi trebala da se koristi, a posljednja adresa je broadcast adresa (kad pošaljemo na tu adresu, svi će čvorovi dobiti pošiljku).
-Na broj adresa moramo dodati još 2 adrese.
/* Pogledati 37, 38 i 39. Slide */

Šta je podrazumijevana putanja?

-Računar komunicira sa čvorovima koji su u njegovoj lokalnoj mreži (što zaključuje na osnovu svoje IP adrese, subnet maske i IP adrese drugog računara). Ako je neki čvor u njegovoj lokalnoj mreži, može sa njim direktno komunicirati. Ako nije u njegovoj lokalnoj mreži, onda mora negdje da pošalje.
-Default route, gateway je uređaj, najčešće router, putem kojeg računar komunicira sa ostalim računarima koji nisu u njegovoj mreži.

Podešavanje mrežnog adaptera:

-Postoje 4 parametra: IP adresa, subnet maska, adresa default gatewaya i DNS (prevodi domenska imena u IP adrese, IP adresa na kojoj se nalazi DNS server).
-Podešavanje može biti kroz GUI ili kroz komandnu liniju
-Ručno se ne podešava zbog DHCP koji omogućava da računar dobije adresu.

Računar pošalje paket na mrežu (na broadcast adresu svima) i zahtjevat će IP adresu i sve ostalo. Ako postoji neko ko mu može to obezbijediti, računar će dobiti adresu iz svoje lokalne mreže, podmrežnu masku, IP adresu default gateway-a i adresu DNS servera. U kućnim mrežama DHCP server je najčešće router.
/* U projektu DHCP je win server */

Address Resolution Protocol (ARP)

-Komunikacija unutar lokalne mreže se odvija preko žice. Dva računara ne mogu jedan drugom poslati paket koristeći IP adrese. Moraju koristiti data link sloj, što znači da bi računar mogao poslati nešto drugom računaru, nije dovoljno da zna IP adresu odredišta, već treba znati njegovu MAC adresu.

Komunikacija između dva računara u istoj podmreži

Ako računar A želi da pošalje paket računaru B, prvo mora da odredi da li je računar B u njegovoj podmreži. To će uraditi na način da izvrši operaciju AND nad svojom IP adresom i podmrežnom maskom, a zatim će istu operaciju primijeniti nad IP adresom računara B i podmrežnom maskom. Ukoliko se dobije ista adresa (adresa podmreže), to je znak da su u istoj podmreži i da oni mogu direktno komunicirati. Nakon toga, računar A šalje ARP paket (broadcast paket), na drugom sloju i kaže da ima IP adresu primaoca, te da mu treba njegova MAC adresa. Računar koji ima tu IP adresu se javi i pošalje svoju MAC adresu.
Svaki računar ima svoju MAC adresu, ona je upržena u mrežni adapter (mrežnu karticu).
Komunikacija između dva računara u istom LAN-u se odvija putem switch-a (uređaj drugog sloja, koji poznaje samo MAC adrese).
-Router ne zna ništa (ništa ne pamti). Zna samo svoju IP adresu i podmrežnu masku.
-Jedno vrijeme se pamte te kombinacije IP i MAC adrese, ali one se ne pamte dugo jer određenu IP adresu može dobiti neko drugi.

-Postoji i Reverse ARP - MAC to IP
-Odgovor na ARP (pitanje “Ko ima ovu IP adresu?”) može neko drugi uraditi - ARP poisoning.

Prosljeđivanje paketa između računara koji nisu u istoj mreži

-Kad se prosljeđuju paketi između računara koji nisu u istoj mreži, pošiljaoc prvo provjerava IP adresu primaoca. Nakon što zaključi da odredišni računar nije u njegovoj lokalnoj mreži, pošiljaoc šalje paket na default gateway. Default gateway mora znati da taj paket nije za njega. Pošiljaoc mora omogućiti da se u paketu nalaze 2 adrese - adresa konačnog odredišta i adresa routera koja će omogućiti da se to lokalno proslijedi.
-Kako se zna IP adresa routera(tj. default gatewaya), mora se poslati upit ARP-u i zatražiti MAC adresa routera. Nakon što dobije MAC adresu, paket se šalje na način da se navede MAC adresa pošiljaoca, MAC adresa primaoca. Za slanje paketa, koristi se switch, pa switch šalje routeru.
-Router će znati da paket nije za njega tako što će raspakovati paket i vidjeti da nije za njega, tj. vidjet će IP adresu konačnog odredišta. Router mora da provjeri da li je primaoc u njegovoj lokalnoj mreži. Ako jeste, može mu poslati direktno (Poslaće ARP zahtjev, dobiti MAC adresu i putem switcha će poslati krajnjem odredištu).
U suprotnom, poslao bi drugom routeru (koristeći IP adresu), konsultujući tabelu prosljeđivanja (rutiranja).
Kako ruter sazna na koji ruter treba poslati nešto? Koristeći algoritme rutiranja.
-Data link sloj se brine samo za lokalnu isporuku!
-Mrežni sloj se oslanja na usluge data link sloja.
-Prosljeđivanje se naziva routing ili rutiranje. Postoje algoritmi koji mogu unutar lokalne mreže ustanoviti najbolju putanju(najkraća, najniže cijene), kako lokalnu tako i globalnu. Globalna se utvrđuje između autonomnih sistema. Unutar jednog ISP-a, ima svoje rutiranje.

Transportni sloj:

-Web preglednik hoće da komunicira sa serverom. Komunikacija se odvija tako što se napravi GET zahtjev koji se upakuje, te dobije IP adresu, koja omogućava da sa računara pošiljaoca dođe do Google-ovog web servera. Kad dođe na Google-ov web server, odredište je aplikacija (nije server). Kako na tom serveru mogu postojati različite aplikacije, mora se znati kojoj od aplikacija pripada taj paket.
-Mrežni sloj omogućava da paket dođe od izvorišta do odredišta. Kad aplikacija šalje paket, preda ga transportnom sloju, on uradi stvari koje će omogućiti da se prepozna koja aplikacija je poslala i koja aplikacija je na odredištu. Onda transportni sloj da IP adresu mrežnom sloju, koji to zapakuje i šalje. Odredišni računar raspakuje paket mrežnog sloja i nađe transportni sloj. Tu se nalazi port. Portovi su u suštini adrese aplikacija na računarima.
-Paket nije završio svoju putanju kad je došao do računara, on završava kad dođe do aplikacije, a to mu omogućava port.
Da bi se omogućilo računarima unutar LAN-a da prime neki paket, koristi se NAT (Network Address Translation). NAT zapravo pretvara unutrašnje IP adrese u vanjske IP adrese. Ideja je bila da bi se uštedile javne IP adrese. Npr. ETF bi mogao imati 1 javnu IP adresu.
Mreža koja ima NAT izvana se vidi samo kao jedna IP adresa, unutrašnja organizacija se ne vidi.
Treba omogućiti da svaki uređaj iz lokalne mreže može komuniciarti sa Internetom i da pritom zadrži svoju privatnu IP adresu.
Glavna motivacija - korištenje IP adrese za cijelu lokalnu mrežu (ušteda)
Ako promijenimo providera, promijenit će se samo javna IP adresa.
Uređaj iz mreže sa adresom 10.0.0.1 (on je izvorište) šalje paket. Kad se pravi paket, odredišni port je adresa aplikacije na odredišnom računaru (uglavnom 80). Izvorišni port je slučajno generisani port, veći od 24. On služi da kad se paket vrati, da se vrati na ispravno mjesto (aplikaciji). Paket dođe do rutera koji ima i NAT funkcionalnosti . Ruter mora promijeniti izvorišnu adresu, umjesto adrese pošiljaoca, on stavi svoju adresu (javnu). Kad se paket vrati nazad, ruter mora znati ko je zaista poslao. On kod sebe mora da zapiše ko je stvarni pošiljaoc paketa (mora znati da je računaru koji je došao sa neke IP adrese i sa nekog porta dodijelio drugu, tj. svoju IP adresu i svoj broj porta). Svaki paket koji prođe kroz NAT će imati istu IP adresu pošiljaoca, ali će broj porta biti različit.
Pitanje: Ako neko iz vanjske mreže želi nekom računaru iz privatne mreže, kome šalje?
Postavke su da ne može. Na uređaju (sa NAT funkcionalnostima) se mora podesiti da mu se može izvana pristupiti.
Ako paket dođe na javnu IP adresu, na neki port, potrebno ga je proslijediti na neku unutrašnju adresu, na neki port - Port forwarding
IPv4 - problem je što je broj adresa (oko 4 milijarde) mal. Broj uređaja koje se konektuju na internet je ogroman.
-Podešava sve na sistemu, uključujući korisnike, pravo upotrebe itd.
-Da li su potrebni neprivilegovani korisnici?
Nisu neophodni, ali potrebni jesu. Redovna upotreba računara treba da bude pod prijavom neprivilegovanog korisnika. Otežava se situacija da zbog greške neprivilegovani korisnik

uradi sistemu nešto što ne bi trebao.
-Nije dobro prijaviti se kao privilegovani korisnik zato što se sve greške (slučajne ili namjerne) su izvršene pod maksimalnim privilegijama, one mogu promijeniti sve na sistemu.
-Kakva je razlika između javnih i privatnih adresa?
Privatne adrese nisu jedinstvene, mogu se koristiti na više mjesta i ne mogu se rutirati.
Javne IP adrese su jedinstvene i sa tim adresama se može direktno komunicirati, mogu se rutirati.
-Parametri na mrežnom adapteru:
IP adresa služi da bude adresa pošiljaoca drugih računara.
IP adresa defaultnog gatewaya - adresa na koju se šalju svi paketi računarima koji nisu u istom LAN-u kao izvorišni računar.
IP adresa DNS servera - IP adresa na koju se šalju pretvaranja IP adresa u domenska imena.
Subnet maska - na osnovu nje može utvrditi sa kojim uređajima ima direktnu komunikaciju.
-Za DHCP su isti parametri.
-Sve se nalazi jedno unutar drugog - hijerarhijski filesystem.
-Računar pošalje paket koji je iza NAT-a, sa svoje privatne IP adrese. Paket na Internetu dobije IP adresu NAT uređaja, kad se paket vraća dođe do adrese NAT uređaja i on pogleda u tabeli na koji port treba proslijediti paket.
-Uređaj koji povezuje podmreže je ruter.
-Kojoj od podmreža pripada? - pripada svim mrežama koje povezuje.
-Podmrežom se smatraju svi računaru koji mogu direktno komunicirati, bez posredstva rutera. Definiše se pomoću subnet maske.
-Kako se ostvaruje da paket poslan sa jednog računara ne bude poslan svim računarima na Internetu (s obzirom na broadcast prirodu slanja na data link slouju)? Kako takav paket ipak dođe do svakog odredišta koje može biti na drugom kraju svijeta? - Na data link sloju, paket se pošalje svima, ali samo oni koji imaju neku MAC adresu pročitaju (data link radi broadcast). Kad paket se pošalje, on ne ode svima.
-Šta znači da je subnet maska 255.255.0.0 - svi računari koji imaju dva okteta ista su u istoj podmreži.
-Treba voditi računa o brzini i vrstama memorije, kako bismo mogli da stavimo to u svoj sistem.

Predavanje 3 – Računari

Životni ciklus računara

1. Novi računar sa instaliranim operativnim sistemom
2. Inicijalizacija sistema, odnosno instaliranje potrebnog software-a
3. Računar dospjeva u konfigurisano stanje
4. Pojava entropije – Prelazak iz konfigurisanog u nepoznato stanje

Obično računar dospjeva u nepoznato stanje nakon određenog vremenskog perioda korištenja istog, odnosno instalacijom sumnjivog software-a.

5. Debugiranjem pokušavamo vratiti sistem u konfigurisano stanje
6. Povremeno potrebno je reinstalirati operativni sistem
7. Kroz razne vrste update-a, vrši se ažuriranje računara.

Cilj jeste da se računar stalno nalazi u konfigurisanom stanju, odnosno da je instalirana najnovija stabilna verzija operativnog sistema, te skup aplikacija koje su korisniku neophodne za rad.

Serveri i personalni računari

Moguće je izvršiti podjelu računara na dvije skupine:

- Namijenjeni za upotrebu jednog korisnika
- Namijenjeni za opsluživanje većine korisnika

Ukoliko ne radi računar od kojeg zavisi jedan korisnik, probleme koje donosi takva okolnost imaće samo taj korisnik. Znatno veći problem predstavlja situacija kada ne radi računar od kojeg zavisi više korisnika.
Računare iz druge skupine nazivamo **serverima** . Oni bi trebali imati veću pouzdanost rada u odnosu na računare iz prve skupine.
U principu svaki računar može obavljati funkciju servera, ali u praksi takvi računari imaju drugačiju hardware arhitekturu, jer PC-ijevi nisu namijenjeni da rade konstantno tokom dužeg vremenskog perioda.

Prodajne grupe

Postoji više prodajnih grupa kome su namijenjeni računari:

- Kućni
- Prodajni
- Serveri

Računari iz prve skupine su većinom jefitiniji od onih namijenjeni prodajnoj grupi, iako imaju iste hardware-ske i software-ske specifikacije. Razlika leži u samim komponentama, koje običnom kupcu nisu važne, dok su za prodajnu grupu veoma bitni. Poslovnim kupcima je najvažnije da računari koje kupuju budu u svakom trenutku u stanju izvršavati zadate

zadatke. Iz tog razloga oni obično naručuju veću količinu računara, gdje imaju mogućnost specifikacije komponenata istog. U slučaju kada otkaže jedna komponenta, korisnici bi brzo mogli izmijeniti istu sa rezervnom komponentom. Time opada ukupan **TCO** (total cost of ownership) te je niži za poslovnog kupca.

Serveri

Server nije računar već program ili proces koji se izvršava na računaru i pruža neku uslugu.

Serverski računari trebaju biti proširivi, odnosno prilikom kupnje takvog uređaja kupac planira korištenje istog na duži vremenski period (do pet godina). Zbog toga prostor namijenjen serverskom računaru moraju imati mogućnost proširenja hardware-skih komponenti.

Procesori imaju bolje performanse od običnih PC-ijeva.

Ulazno/izlazni uređaji su jako bitni kod serverskih računara, jer oni konstantno komuniciraju između sebe. Zbog toga poboljšane performanse U/I uređaja su prioritet za serverske računare.

Ovi računari su nadogradivi, odnosno većinom ne predstavljaju „stand alone“ uređaje već se nalaze u tzv. **rack** -ovima.

Serveri imaju rezervne komponente (više naponskih jedinica, rezervni diksovi), koje omugućavaju da u slučaju kvara komponente, ista budu zamijenjena a da se server ne isključuje.

Također, održavanje ovakvih računara je bolje, odnosno garantirano je na najmanje tri godine.

Većina servera ima omogućeno daljinsko upravljanje. To podrazumijeva da je kroz mrežu moguće pristupiti hardware-u servera (nezavisno od operativnog sistema). To je obezbijeđeno ugradnjom određenih mrežnih kartica.

Alternativni serverski hardware

U prošlosti filozofija na kome su bili bazirani serverski sistemi jeste da jedan hardware-ski server ima jednu software-sku namjenu, tj. jedna fizička mašina je mail server, jedna fizička mašina je web server.

U zadnje vrijeme došlo je do promjene u takvom vidu izgradnje serverskih sistema. Jedna ideja jeste da više fizičkih mašina (jeftnijih, jednostavnijih računara) obavlja jednu funkciju. Na takvom principu su bazirani **blade serveri** , čiji hard diskovi su fizički odvojeni od memorije i procesora. To nazivamo **SAN** (storage area network), koji je vezan za pohranjivanje podataka.

Danas se najčešće koristi veliki server na kome se nalazi veći broj manjih blade servera , koji zapravo predstavlja jednu računarsku platformu na kome se instaliraju virtuelne mašine koje čine virtualne servere.

Server soba

Prostorija gdje se nalazi oprema potrebna za rad i održavanje serverskih sistema, tj. nalazi se serverska i komunikaciona infrastruktura sistema.

Pristup server sobi trebaju imati samo određene osobe, odnosno potrebno je maksimalno ograničiti pristup istoj.

Neophodno je bezprekidno napajanje, jer takvi uređaji bi trebali biti stalno u funkciji. Korištenjem **UPS** -a (uninterruptible power supply) omogućeno je da u slučaju prestanka električnog napajanja, serveri nastave raditi određeni vremenski period, dovoljan za pohranu podataka.

Server soba mora biti klimatizovana, odnosno neophodno je obezbijediti ambijentnu temperaturu u sobi, kako ne bi došlo do iznenadnog kvara na serveskim komponentama.

Mora biti obezbijeđena redudantnost, odnosno da postoji više komponenti nego što je neophodno za minimalno funkcionsanje servera. Kabliranje treba da je provedeno, tj. da za svaku konekciju postoji više kablova. Potrebno je da postoji više računara nego što je potrebno, kao i da su dostupni diskovi koji se u datom trenutku ne koriste.

Server soba se treba nalaziti u antistatičkom okruženju, jer su elektrosnke komponente osjetljive na statički kapacitet.

Za neometani rad u server sobi potrebno je omogućiti lak pristup opremi i kablovima.

Oprema u server sobi treba biti obilježena i dokumentovana, jer ukoliko dođe do promjene administratora, zamjena istog treba se lahko i brzo snaći u novom okruženju.

Server soba treba biti zaštićena od elementarnih nepogoda, kao što su poplave, udar groma, zemotrljes itd. Treba izbjegavati gradnju servera na nesigurnim lokacijama, npr. u prizemlju stambenog objekta.

Backup podataka se ne bi trebao nalaziti na istoj fizičkoj lokaciji kao i originalni zapis podataka da bi se spriječio mogući nestanak svih kopija podataka sistema u slučaju krađe, terorističkog napada ili elementarne nepogode.

Fizička lokacija server/a ne bi trebala biti upečatljivo naznačena budući da može privući samo neželjenu pažnju (tipa da banka lokaciju, gdje se nalaze njeni serveri, na ulazu označi ogromnim bliještečim naslovom).

Podjela diska na djelove

Svaki disk, tj. njegov fizički prostor se može podijeliti na više dijelova, što nazivamo **particionisanje diska,** pri čemu se particije ne preklapaju. Cilj particionisanja jeste odvajanje file-ova za različite namjene.
Dobra je praksa operativni sistem držati na jednoj particiji, a korisničke file-ove na drugu. Ukoliko se disk ne podijeli na dio rezervisan za operativni sistem i dio namijenjen korisničkim podacima, u jednom trenutku može doći do situacije gdje će se disk prepuniti sa korisničkim podacima, čime bi bio onemogućen normalan rad operativnog sistema računara.
Veličina particije zavisi isključivo od potreba korisnika, odnosno optimalna mjera podjele se postiže kroz planiranje i namjene za šta će se koristiti data particija.

Particioniranje

Postoji više alata: **fdisk** (cfdisk), **gParted** itd.
PC Bios omogućava kreiranja 4 primarne particije diska. Moguće je i kreiranje proširene particije unutar koje je moguće kreiranje dodatnih particija. Primarna particija je particija s koje se može pokrenuti operativni sistem. Kod BSD i Sun-a particionisanje je nešto drugačije.

Formatiranje particija

Podjela diska na particije i formatiranje particije su dvije potpuno različite operacije. Nakon podjele diska na particije, neophodno je uspostaviti datotečni sistem na kreiranoj particiji, što se ostvaruje formatiranjem diska. Datotečni sistem omogućava da se pohranjeni biti mogu interpretirati kao datoteke. Omogućeno je i davanje imena particiji.
Na Windows operativnim sistemima imenovanje particija kreće od slova C do Z.
Kod Linux-a particije se imenuje nešto drugačije. Primjer linux imenovanja particija:
/dev/hda0... /dev/sdb1
Hd i sd se odnose na vrste kontrolera računara. Hd su bili sada već zastarjeli IDE kontroleri, a sd predstavljaju SCSI i SATA diskove. Slovo a u gornjem primjeru se odnosi na redni broj diska (prvi u ovom slučaju). Broj na kraju labele se odnosi na redni broj kreirane particije.
Postoji više različitih alata pomoću kojih je omogućeno formatiranje particija, zavisno od operativnog sistema na kome se vrši formatiranje (format, mkfs, newfs).

Organizacija datotečnog sistema

Prilikom particionisanja treba odlučiti kako organizovati datoteke na računaru. Operativni sistem bi trebao biti pohranjen na posebnoj lokaciji. Kod Windows-a podaci o operativnom sistemu se na čuvaju na posebnoj particiji, već samo u određenom folderu, u ovom slučaju folder „windows“. Software drugih proizvođača se čuva u folderu „Program files“.
Potrebno je odvojiti i prostor za korisničke datoteke, odnosno u slučaju Windows OS-a oni se nalaze u folderu „Users“. Prednost takvog organizovanja podataka jeste da korisnička podešavanja možemo preseliti na drugi računar, a da ona ostanu identična. Privremeni radni prostor na Windowsu se nalazi u „temp“ folderu, koji je često popunjen „junk“ podacima, što može izazvati probleme ukoliko se nalazi na istoj particiji kao i operativni sistem.
Podaci koji se često ažuriraju, trebaju se nalaziti u određenoj datoteci, kako bi kreiranje back-up-a bio što brži i jednostavniji.
Datotečni sistemi su hijerarhijski organizovani, kako bi se mogla kreirati logička struktura podataka.
Inicijalno odvajamo korisničke podatke, proizvode koje trenutno razvijamo, kao i lokalne baze podataka.

Instalacija operativnog sistema

Prilikom instalacije korisnik treba paziti kakve komponente želi instalirati, jer ono što je potrebno korisniku nije nužno identično onome što instalira operativni sistem.
Kakvu instalaciju izabrati zavisi prvenstveno od namjene računara, odnosno da li korisnik želi PC ili server. Kao primjer možemo navesti verzije Windows OS-a koji u većini funkcija identični, ali se razlikuju u namjeni za koga je operativni sistem kreiran (Windows server 2016 i Windows 10).
Računaru je potrebno dati: ime, IP adresu i podmrežnu masku, ime domene, lokalnu vremensku zonu, kao i da se odredi veličina virtuelne memorije (na Linux OS-u to predstavlja **swap space**).

Više operativnih sistema na jednom računaru

Svaki operativni sistem treba imati rezervisanu svoju particiju.
Boot manager predstavlja software koji se pokreće na disku i nudi mogućnost odabira pokretanja jednog od operativnih sistema koji su instalirani na datom disku.

Kloniranje

Ukoliko postoji veći skup računara koji ima iste hardware-ske komponente, možemo izvršiti instalaciju identičnu software-sku konfiguraciju. U prošlosti su se koristili CD-ovi pomoću kojeg bi na svakom posebno instalirali dati software, te samo promijenili IP adresu. Danas su takvi procesi dosta ubrzani koristeći jednu od mnogobrojnih mrežnih usluga, tako što se odabire jedan računar koji će predstavljati **master** uređaj, dok će ostali obavljati funkciju **slave** jedinica. Na master računaru prvo bi se instalirao odgovarajući software, a zatim bi na ostalim računarima pokretao cd, koji bi se boot-anjem na mreži potražio server (dati master računar) koji će slave uređajima preko mreže poslati instalacione datoteke.

Software licenciranje

Prilikom instaliranja software-a potrebno je paziti i na licencu koji sa sobom. Ona reguliše pravo korištenja i mijenjanja. Može postojati striktna kontrola svega od strane proizvođača, pravo korištenja, ali ne i mijenjanja software-a, te pravo korištenja i mijenjanja.

Open source licence omogućavaju korisniku pristup izvornom kodu, ali ne znače nužno da imaju pravo i mijenjanja tog koda.

Firme koje prave aplikativni software, nerado daju pristup izvornom kodu, jer time se izlažu opasnosti da će klijenti razviti software sličan njihovom i komercijalizirati isti.

Postoje licence u kojim se navodi da je software strogo otvoren, odnosno ukoliko korisnik iskoristi izvorni kod iz datog software-a za kreiranje svog projekta, prilikom publikacije istog on također mora biti tipa open source.

Instalacija software-a se može vršiti iz određene izvršene verzije i iz izvornog koda.

Kod Unix-a konfiguracione datoteke se spašavaju u folderu /etc. Pri tome one predstavljaju tekstualne datoteke.

Instalacija iz paketa i izvornog koda – Linux

Paket predstavlja software zapakovan na određen način, koji uključuje sve biblioteke potrebne za instalaciju. Ovo je najčešći način instalacije software-a na Linux OS. Alternativu predstavlja instalacija preko izvornog koda. To je najčešće slučaj kada izađe nova verzija software-a koja je korisniku neophodna za daljni rad, a pri tome nije izašao paket sa novom verzijom software-a.

Instalacija iz izvornog koda se sastoji iz tri dijela. Prvi korak jeste unos komande „./configure“ (ovim postupkom osiguravamo se da će instalacija biti pokrenuta iz lokalnog direktorija (zbog toga tačka)). Komanda „configure“ zapravo priprema (generiše) **makefile** . Ova datoteka predstavlja skup instrukcija koje služe compiler-u i linkeru da iz izvornog koda naprave izvršni kod.

Komanda **make install** omogućava da se taj program instalira na način da bude dostupan svim korisnicima.

Instalacija software-a na mobilnim uređajima

Instalacija je veoma slična instalaciji paketa na Linux OS-u, pri čemu kod Androida i iOS-a korisnik preuzima paket u kome se nalazi sav software potreban za uspješnu instalaciju aplikacije. Paketi se mogu preuzeti sa oficijelnih repozitorija (Playstore, Appstore itd) ili direktno preuzimanje paketa sa alternativnih stranica (što nije preporučljivo).

Pokretanje računara

BIOS predstavlja komad software-a koji se nalazi na matičnoj ploči računara. Ne nalazi se na disku, jer se on pokreće nezavisno od svih ostalih komponenti računara. BIOS ima dvije funkcije:

Kao prvo vrši provjeru hardware-a računara, odnosno POST (Power on Self Test) – Tokom tog testa BIOS šalje signale na sabirnici različitim uređajima, te čeka odgovor od uređaja. Vrši se provjera sa konfiguracijskom listom u kojoj se nalaze podaci o prethodno priključenim uređajima na matičnu ploču, ukoliko jedan ili više uređaja nije poslao povratni signal, BIOS javlja grešku i obavještava korisnika o tom problemu (ukoliko je to moguće poruka se ispisuje na ekranu računara).U slučaju da nije moguće ispisat grešku, BIOS upozorava korisnika zvučnim signalima.

Mada je POST veoma siguran način testiranja hardware-a računara, on neće detektovati grešku koja je nastala na hard disku. Tek nakon pokretanja OS-a prilikom pokušaja pristupa određenim datotekama na defektnom disku doći će do ispisa greške.

Ukoliko POST prođe uspješno, BIOS počinje sa izvršenjem druge svoje funkcije, tj. pronalaska medija (hard disk, floppy disk, USB itd.) na kome se nalazi operativni sistem. Prilikom podešavanja BIOS-a moguće je odrediti kojim redoslijedom će se provjeravati medij da bi se pokrenuo OS, tzv. **Boot loader.**

Nakon što je pronašao medij na kome se nalazi OS, BIOS pokreće software sa medija koji se naziva **Boot manager** . Preko njega se vrši pokretanje OS-a.

Pokretanje OS - Unix

Init je prvi proces koji se pokreće na Unix OS-u i on predstavlja parent proces ostalim procesima u OS-u. U konfirguraciji operativnog sistema postoji tzv. **Run level** , koji definiše programe koji će se pokrenuti. Prilikom odabira jednog od run level-a, bit će prikazan spisak aplikacija koji će se pokrenuti.

Svrha run level-a jeste da korisnik može odabrati skup aplikacija koji će se pokrenuti prilikom podizanja OS-a, odnosno sve zavisi od korisničkih potreba i namjera.

Na UNIX-u postoji **jednokorisnički način rada** . Ukoliko korisnik želi da vrši jednu vrstu sistemskog održavanja, pri tome računar ne bi trebali koristiti ostali prijavljeni korisnici, dati korisnik pokreće ovaj način rada (npr. kad se vrši kloniranje).

Pokretanje OS – Windows

Na Windows-u se pokreće **Windows Boot manager** , odnosno na starijim verzijama **NTLDR** . U registrima piše spisak programa koji će se pokrenuti prilikom podizanja OS-a. Prednost ovakvog pristupa jeste da ukoliko je na računaru instaliran zlonamjeran software, isti će najvjerovatnije biti zapisan u registrima, odnosno putanja datoteke iz koje se pokreće

isti, te je moguće poslije odstraniti dati malware.
Na Windowsu ne postoji tradicionalni jednokorisnički način rada, ali postoji tzv. **Safe mode** . Na računaru se pokreće minimalan skup driver-a i procesa neophodno za podizanje OS-a. Ideja je takva da ukoliko je došlo do neočekivanog ponašanja računara, da će se moći pronaći uzrok toga u safe mode-u, jer da je kvar nastao na hardware-u, BIOS bi obavijestio korisnika već tokom POST-a.

Izbor programa koji se pokreću

Većina programa koji se pokreću na OS-u predstavljaju **Services** , odnosno servise koji pružaju određene usluge.
Servisi po tipu **Startup Type** mogu biti: **automatic** , **manual** i **disabled** .
Automatic servisi se pokreću prilikom pokretanja OS-a. Manual servici se pokreću samo onda kada je potreban Windowsu ili nekom drugom servisu, te ukoliko korisnik pokrene određeni proces koji pokreće dati proces. Disabled servisi su takvi servisi koji će biti isključeni, bez obzira da li korisnik želi da ga pokrene. Izvršavanje ostalih servisa ili aplikacija koje zavise od isključenog servisa će vjerovatno biti neuspješno.
Na UNIX-u postoji runlevel editor koji posjeduje grafičko okruženje i konfiguracione datoteke, gdje korisnik može odabrati koji servisi će se pokrenuti a koji ne.
Servisi predstavlja proces na računaru, ali svaki proces nije ujedno i servis. Servisi su procesi koje pružaju usluge drugim procesima.

Pokrenuti procesi – Ubuntu

Komanda **ps** na Ubuntu-u omogućava pregled pokrenutih procesa na računaru. Komanda **top** prikazuje koja aplikacija, odnosno proces zauzima najviše resursa na računaru.
Netstat komanda prikazuje mrežene usluge koje pruža računar, odnosno mrežne konekcije na TCP-u, kao i tabele rutiranja i mrežne interfejse na računaru.

Pokrenuti procesi – Windows

Na Windows-u pokrenute procese možemo posmatrati pomoću Windows Defender-a, pri čemu postoji više kategorija prikaza takvih procesa: Oni koji se pokreću na Start up-u računara, programe koji uspostavljeni uz mrežnu konekciju kao i trenutno pokrenute procese.

Gašenje računara

Mehanički uređaji (npr. hard disk) trebali bi se uredno isključiti, kako ne bi došlo do njihovog kvara. Svi procesi koji su u datom trenutnku aktivni trebaju biti završeni, prije nego što se vrši isključivanje uređaja. Zapis na diskove treba biti okončan, inače može doći nepopravljive štete na istom (npr. isključivanje USB sticka iz računara, iako proces prijenosa podataka na USB nije završen).
Na Linux OS-u komandom **shutdown** , te unosom dodatnih parametara (npr. –h +5 -sistem se gasi za 5 minuta) pokrećemo proces gašenja računara.
Na verzijama OS-a koji je namijenjen za radnju sa serverima prilikom gašenja računara biva prikazan meni na kome treba odabrati razlog zbog kojeg se gasi računar. Takvim pristupom korisnik će prilikom sljedećeg paljenja računara biti obaviješten zašto je došlo do gašenja servera.

Pitanja za ovo predavanje:

1. Šta je kloniranje i kada i zašto je potrebno?
2. Objasniti događaje koji se odvijaju od paljenja računara do pokretanja operativnog sistema.
3. Da li je moguće instalirati softver iz izvornog koda na Windows OS i ako jeste šte je za to potrebno?
4. Kako organizacija datotečnog sistema može olakšati pravljenje sigurnosnih kopija (*backup*)?
5. Ako je potrebno instalirati softver na Linux OS predložite koji od načina instalacije bi ste izabrali i zašto.
6. Šta je BIOS i gdje se nalazi?
7. U čemu se razlikuju procedure pokretanja Windows i Unix OS?
8. Objasniti razliku između pravljenja i formatiranja particija.
9. Koja je uobičajena lokacija za instalaciju softvera na Windows OS? Može li se softver instalirati na neku drugu lokaciju? Koje su prednosti i nedostaci instalacije softvera na lokaciju drugačiju od uobičajene?
10. Šta je *run level* kod Unix-oidnih OS?
11. Šta su „services“ u Windows OS? U čemu je razlika ako je za neki servis za način pokretanja navedeno: *automatic, manual* ili *disabled* ?
12. Na kojim medijima i kojim redom BIOS traži OS koji će pokrenuti?
13. Na koju fizičku particiju kog diska se odnosi logičko ime particije /dev/sdb2?
14. Na koju fizičku particiju kog diska se odnosi logičko ime particije /dev/sdc1?
15. Šta je BIOS? Gdje se nalazi i koja mu je funkcija?
16. Na primjeru hard diskova objasniti šta bi značilo da su oni u redundantnoj konfiguraciji.
17. Na koju lokaciju se instalira softver na Unix-oidnin OS? Na koju lokaciju je bio

instaliran softver koji ste koristili pri realizaciji projekta?
18. Objasniti razliku između boot loadera i boot sektora.

Predavanje 4 – Virtuelizacija i analiza mreže

Virtuelizacija

Pod pojmom virtuelizacije mislimo na stvaranje prividne stvarnosti, odnosno virtuelne realnosti.

Virtuelizaciju možemo definisati kao skup raznih tehnologija namijenjene za upravljanje računarskim resursima, stvarajući tako abstraktni sloj između software-a i fizičkog hardware-a. Te tehnologije efektno emuliraju ili simuliraju hardware-ske platforme (najprije servere, uređaje za skladištenje podataka, mrežne resurse itd.) u obliku određenog software-a.

Kao što keš omogućava brzi pristup dijelovima programa i podataka iz memorije koji se najčešće koriste, tako i memorija može "keširati" dijelove diska koji se često koriste. Ova tehnika se naziva **virtuelna memorija** i ima zadatak da programeru (i njegovom programu) da iluziju da mu je na raspolaganju proširena (praktično neograničena) radna memorija, zaštićena od pristupa drugih korisničkih programa.

Virtuelna memorija se najčešće oslanja na diskove, kao sljedeći niži nivo u memorijskoj hijerarhiji, jer zadržavaju svoj sadržaj i nakon nestanka napajanja i, po pravilu, imaju mnogo veći kapacitet od fizičke radne memorije (DRAM-a).

Virtuelne privatne mreže daju privid da se računari, koji se nalazi na vanjskoj mreži, predstave kao da se nalaze na privatnoj mreži.

Virtuelna mašina – jedna instanca operativnog sistema zajedno sa jednom ili više aplikacija pokrenuta na izolovanoj particiji unutar računara.

NFV (network functions virtualization) predstavlja virtuelizaciju mrežnih funkcija, tako što se vrši implementacija tih funkcija u software-u, te se vrši njihovo pokretanje na virtuelnoj mašini.

Ideja virtuelizacije

Virtuelizacija je bazirana na apstrakciji. Pokušavaju se fizički resursi pretvoriti u logičke, odnosno virtuelne. Apstrakcija fizičkog hardware-a vrši se kroz određene software-ske interfejse. Primjer apstrakcije jeste datotečni sistem, koje daje korisniku samo privid da on pristupa folderima i datotekama, mada su to samo grupisani biti zapisani na disku. Korisnici, aplikacije i upravljački software ne zanimaju se fizičkim detaljima.

Virtuelizacija mreža je zasnovana na ideji da se kombinuju dostupni fizički mrežni resursi u virtualne koji se dodjelu po potrebi.

Tipičan primjer virtuelizacije hardware-a jesu virtuelne mašine.

Virtuelizacija hardware-a

Ukoliko korisnik razvije jednu aplikaciju koja je bazirana na određenom OS-u, te želi testirati njenu kompatibilnost na drugim operativnim sistemima, da ne postoji virtuelizacija, on bi morao svaki OS posebno instalirati na računaru, jer je moguće izvršavanje samo jednog OS-u datom trenutku.

Ideja virtuelizacijskih tehnologija jeste da korisnik na jednom fizičkom hardware-u može izvršavati više operativnih sistema, odnosno više virutelnih mašina.

Prve korake ka virtuelizaciji napravio je IBM 1970.-ih godina. Tema je aktuelizirana tokom 2000.-ih godina od strane Windows-a jer je došlo do toga da je hardware nadrastao software.

Serverski računari bili su većinu vremena neiskorišteni, jer su se određeni servisi (web server, mail server itd.) implementirali na različitim fizičkim jedinicama. Da bi se postigla maksimalna iskorištenost hardware-a, ponovo je pokrenuta ideja virtuelizacije, gdje bi se na jednom računaru pokretalo više VM-ova na kojim bi se implementirali dati servisi.

Virtuelna mašina

Operativnom sistemu se daje privid hardware-a (fizičke mašine) na kome se on izvršava. OS „zaključuje" da se izvršava direktno na hardware-u.

Komunikaciju između virtuelne mašine i stvarnog hardware-a obavlja komad software-a koji se naziva **monitor virtuelne mašine (hypervisor),** koji omogućava da se na jednoj fizičkoj mašini može nalaziti više virtuelnih, svaka sa svojim OS-om.

Napomena: Ukoliko je na host OS-u instalirana 32 bitna verzija istog, nije moguće podići virtuelnu mašinu s OS-om koji je 64 bitni (obrnuto je moguće).

Između hardware-a i svake od virtuelnih mašina postoji komad software-a koji vrši međusobnu kominukaciju, odnosno dodjeljuje dio fizičkog hardware-a virtuelnim mašinama.

Korisnik može dodijeliti virtuelnim mašina u zbiru više fizičkih resursa nego što ima na raspolaganju, iz razloga što se ti resursi u stvarnosti nikada neće u potpunosti potrošiti.

Konsolidacijski odnos (ratio) daje odgovor koliko VM-ova je moguće imati na jednom fizičkom računaru. Ukoliko je na host-u moguće kreirati 6 VM-ova , onda govorimo o konsolidacijskom odnosu 6:1.

Danas postoji više virtuelnih servera nego fizičkih.

Hardware je postao apstraktan resurs. OS i aplikacije mogu dobiti prividno onoliko hardware-a koliko im treba za funkcionisanje, jer njih ne zanima odakle i kako se dobavlja taj

hardware. Time je puno bolje iskorišten hardware. Proces migracije VM s jednog fizičkog hardware-a na drugi je postao veoma lak i brz. Postoji više razloga za migraciju VM-a sa jednog hardware-a na drugi.
Ukoliko dodje do otkaza hardware-a, korisnik bez problema može prebaciti VM na novi hardware, te će VM i dalje funkcionisati.
Situacije kada postoje više fizičkih servera, od kojih je jedan veoma opterećen VM-ovima, a drugi gotovo neiskorišten, vrlo lako se može riješiti ponovnim raspoređivanjem opterećenja, odnosno migracijom određenih VM-ova s jednog servera na drugi.
Migracija VM-ova na drugi hardware može nastati iz razloga ukoliko su povećane/smanjene potrebe za tom virtuelnom mašinom, odnosno ukoliko je na jednom fizičkom serveru instalirano deset virtuelnih mašina koje maksimalno iskorištavaju fizičke resurse istog, veoma lako se može vršiti premještanje one VM koja nije ključna za funkcionisanje na drugi hardware, kako bi se oslobodili fizički resursi.

Arhitekture

VMM je posrednik (proxy) između VM-a i hardware-a.
Prilikom instaliranje VM-ova doći će do određenog opadanja performansi, jer određeni dio fizičkih resursa će iskoristiti i sam VMM, a ne samo virtuelna mašina.
Virtuelna mašina je zapravo skup datoteka. Ona se najčešće kreira u jednom folderu. U datom folderu se nalazi konfiguracijska datoteka, koja opisuje virtuelni računar (CPU, RAM, HD, NIC, I/O, ...).
Druga veoma bitna datoteka jeste **virtuelni hard disk** , koju VM vidi kao hard disk na virtuelnom računaru. Ova datoteka se nalazi na host računaru, pri tome datotečni sistem datoteke će odgovarati host uređaju, a VM će imati samo privid da ta datoteka posjeduje datotečni sistem koji odgovara VM-u. Primjer: Na Linux host OS-u se kreira VHD za VM Windows 10. Ta datoteka će biti tipa ext4, ali unutar VM-a ovaj disk će se ponašati kao NTFS.
Nakon pokretanja VM-a, dodatne datoteke su kreirane namijenjenje za memory paging, logging i ostale funkcije.
Backup podataka je kritična funkcija u svijetu računara. Kako VM-ovi predstavljaju datoteke, kopiranje istih ne donosi samo backup podataka već i kopiju cijelog servera, uključujući operativni sistem, aplikacije, kao i podatke o podešavanju datog hardware-a.
Da bi se kreirala kopija fizičkog servera, potrebno je nabaviti dodatni hardware, instalirati i podesiti isti, pokrenuti OS, aplikacije prije nego što bi se mogao predati korisnicima, što može potrajati sedmicama ili čak mjesecima zavisno od kompleksnosti procesa.
Kako se VM sastoji iz datoteka, umnožavanjem istih, u virtuelnom okruženju kreirat će se perfektna kopija servera u samo nekoliko minuta. Potrebno je podesiti određene podatke (ime servera i IP adresa), ali administratorima to predstavlja veoma kratak posao (od nekoliko minuta do nekoliko sati) , za razliku od mjeseci koji su potrebni za isti postupak na fizičkom računaru.
Još jedan metod koji se koristi u umnožavanju VM-ova jeste kroz upotrebu **template-a** .
Upotrebom template-a korisnik ima pristup standardiziranoj grupi hardware-a i software-a koji se mogu iskoristiti za podešavanje i kreiranje novih VM-ova (koje će biti biti upravo tako podešene). Primjer: Ukoliko je korisniku potrebno 10 virtuelnih mašina, on to može postići pomoću template-a, koji za njega generise potrebne podatke, tako da se svaka VM razlikuje jedna od druge (IP adresa, ime servera itd).
Iz više razloga virtuelna okruženja su postala novi model za infrastrukturu data centara. Jedan od tih jeste povećana pouzdanost. VM host-ovi su spojeni u zajednički klaster, da bi formirali „bazen“ računarskih resursa. Veći broj VM-ova su host-ani na jednom od tih servera u slučaju da fizički server otkaže, jer se VM-ovi na defektnom serveru mogu brzo premjestiti i automatski restartovati na drugom host-u unutar klastera. Time pružaju veću pouzdanost za drastično manju cijenu i kompleksnost.
Jedan od najzanimljivijih osobina virtuelnih okruženja jeste mogućnost premještanja VM-a koji je pokrenut sa jednog fizičkog host-a na drugi, a pri tome ne dolazi do prekida rada VM-a, te ne utiče na korisnike te virtuelne mašine. Ta osobina se naziva **Live Migration** (**vMotion** unutar VMware okruženja). Ukoliko nema dovoljno hardware-skih resursa bez problema se može izvršiti premještanje VM-a s jednog servera na drugi.

Tipovi hipervizora (VMM)

Hipervizori se nalaze između hardware-a i virtuelne mašine. Postoje dva tipa hipervizora, koji se razlikuju po tome da li se nalazi dodatni OS između hipervizora i host-a.
Prvi tip hipervizora predstavlja „tanki“ software-ski sloj koji se pokreće direktno u fizičkom serveru, kao što se OS pokreće. Kada je instaliran i podešen, u vremenskom periodu od nekoliko minuta, server može pokretati VM-ove. Neki primjeri ovog tipa hipervizora čine **VMware ESXi, Microsoft Hyper-V** itd.
Drugi tip hipervizora predstavlja tradicionalnu aplikaciju, odnosno programski kod koji se pokreće u Windows ili UNIX/Linux okruženju. Neki primjeri drugog tipa su: **VMWare Workstatiton i Oracle VM Virtual Box** .
Prvi tip hipervizora se nalazi direktno na fizičkom host-u i kao takav može kontrolisati fizičke

resurse tog hosta, dok drugi tip hipervizora se nalazi između OS-a i tih fizičkih resursa. Kako se Tip 1 hipervizora ne natječe sa OS-om za resurse, postoji više slobodnih resursa na host-u, prema tome moguće je pokrenuti više virteulnih mašina. Oni se generalno smatraju sigurnijim od hipervizora drugog tipa.
Prednost drugog tipa hipervizora jeste da korisnik ima mogućnost korištenja OS-a koji se pokreće na fizičkom hardware-u za svoje potrebe, te može iskoristiti virtualizacijske tehnologije koje pruža virtuelno okruženje.

Umrežavanje i VM

Tokom umrežavanja virtuelne mašine postoji više načina rada.
Korisnik može odrediti da VM funkcioniše bez mreže, odnosno da nije umrežena.
NAT način rada (pretpostavka jeste da je host priključen na switch). VM će se ponašati kao da je host zapravo NAT uređaj, odnosno da će virtuelna mašina dobiti IP adresu od tog NAT uređaja.
Premošteni način rada – kao da je VM priključena u isti switch kao i host (host može ping-at VM, VM može ping-at host, tj. mogu međusobno komunicirati).
Interna mreža – možemo podesiti da dvije ili više virtuelnih mašina budu prividno priključene u isti switch, odnosno čine jednu mrežu, čime je uspostavljena međusobna komunikacija.
Mreža samo sa monitorom VM (Host – only) – VM je priključena na switch sa hostom, odnosno izolirana je od vanjske mreže. Ovaj način rada predstavlja internu mrežu sa host-om.

Virtuelizacija kontejnerima

Software poznat kao **virtuelizacijski kontejner** , pokreće se na vrhu host OS kernel-a i omogućava izvršno okruženje za aplikacije. Za razliku od virtuelnih mašina bazirane na hipervizorima, kontejneri ne pokušavaju emulirati fizičke servere. Sve aplikacije koje se nalaze u kontejneru na hostu dijeli zajednički OS kernel. Ovakvim pristupom nisu potrebni posebni resursi za svaku aplikaciju kako bi se pokrenuo OS za svaku aplikaciju.

NFV

Izvedba mrežnih funkcija u software-u u VM-u. U tradicionalnim mrežama, svi uređaji su izgrađeni na zatvorenim platformama. Svi mrežni elementi su ograđene kutije, i hardware ne može biti dijeljen. Svaki uređaj zahtijeva dodatni hardware za povećani kapacitet, ali taj hardware je besposlen kada sistem radi u smanjenom kapacitetu. Sa NFV-om svi mrežni elementi su nezavisne aplikacije koje su fleksibilno izgrađena na ujedinjenoj platformi, koja se sastoji iz standardnih servera, uređaja za pohranu podataka i switch-eva.

Zajednica

Umreženi računari predstavljaju jednu vrstu zajednice. U kooperativnim zajednicama ono što pojedinac radi utiče na sve članove zajednice stoga je svaki pojedinac odgovoran da se ponaša odgovorno prema drugim članovima zajednice.
U bilo kakvoj zajednici, postoje pravila i zakoni koji se moraju poštovati.
Sistemska politika predstavlja izjavu o ciljevima i željama koja je što je moguće više iskazana kroz formalne projekte infrastrukture i šemu odgovornosti i ponašanja za moguće događaje.
Uniformnost (**hardware-a** i **software-a**) – primjer: kupovina većeg broja istih računara (isti hardware i software) , prednost takvog pristupa jeste u manjoj razlici i odstupanjima o kojim treba voditi računa, te je veća statička predvidljivost, odnosno ukoliko se jedan računar pokvari, nije teško zamijeniti isti. Nedostatak leži u činjenici da su određeni hardware-ski uređaji podobniji kvaru. Primjer: postoji verzija hard diskova koja nakon nekog vremena počinje otkazivat, pri tome će svi uređaji biti pogođeni tim kvarom, te je potrebno vršiti izmijenu hardware-skog uređaja na cijelom sistemu.
Raznolikost – vrši se raspoređivanje rizika od mogućih problema sa pojedinim komponentama. Isti problem uglavnom en pogađa različite komponente.

Mrežni modeli

Mainframe – jedan centralni računar na kome se vršilo svo procesiranje podataka. Svi klijentski računari su isključivo terminali.
Unix distribuirani model - da svaki Unix računar može ujedno biti i server i kljient.
Windows centralizirani model – Windows je jasno podijelio operativne sisteme na klijentske i serverske. Ideja je bila da se sve mrežne usluge koje se pružaju, izvršavaju na serverskim operativnim sistema.
Macintosh - uveo Apple GUI, OS X (BSD) , TCP/IP (kombinacija windows i unix modela)
Novell Netware – oni su prvi omogućili umrežavanje
Peer to peer - mreže zasnovani na distrubiranom modelu , svi učesnici u komunikaciju su ujedno i klijenti i serveri (primjer: Torrenti – korisnik koji skida torrent je u tom slučaju klijent, dok *seed* -anjem se ponaša kao server, jer pruža drugima određenu uslugu)
Cloud computing – praksa da se pohrana, upravljanje i procesiranje podataka vrši na mreži udaljenih servera na internetu, a ne na lokalnom serveru ili personalnom računaru.

Mrežni servisi

To su usluge koje se pružaju korisnicima mreže.
Server nije računar već program ili proces koji se izvršava na računaru i pruža neku uslugu.

(spomenuto u predavanju 3).
U Windows okruženju servis možemo naći pod imenom **service** , dok u Unix-u se naziva **deamon** (jer se izvršavaju procesi u pozadini, nevidljivi su, odsluškuju na određenom mrežnom portu i ako dobiju zahtjev s mreže odgovaraju na iste).

Pohranjivanje korisničkih podešenja

Analiza mreže

Postoje dva plana prilikom analize: izgradnja nove mreže (bolja varijanta, ali ne tako česta u praksi), te analiza postojeće mreže.
Na postojećoj mreži je neophodno odrediti od čega se ona sastoji, tj. koji se resursi nalaze u toj mreži.
Topologija – kako su čvorovi povezani, odnosno kako je mreža fizički spojena.
Spisak računara i ostalih hardware-skih uređaja unutar mreže, gdje se nalaze itd.
IP adresiranje – koliko ima podmreža, koje su njihove adrese, postoje li ruteri i koliko ih ima u mreži i koje su podrazumjevane rute
Lokacija ključnih mrežnih servisa – gdje se nalazi web server, mail server, dns, active directory, gdje su file serveri na kome se pohranjuju određeni podaci itd.
Potrebno je znati koliko ukupno ima računara u mreži, te kako se svaki od njih zove i koja mu je adresa. Također, bitno je znati kakva je hardware-ska konfiguracija računara, odnosno kakvi su im procesori, memorija, diskovi, itd. Važna informacija jeste kakav je OS instaliran na računaru.
Kabliranje – gdje su sprovedeni kablovi, koliko ih ukupno ima, koliko ima utičnica ukupno za svaki kabal
Komunikacioni ormarići – gdje se nalaze, koliko ih ima...
Mrežna oprema – koju vrstu mrežne opreme možemo naći u mreži, koliko ima switch-eva i hubova koji predstavljaju **koncentratore** . Potrebno je znati postoje li ruteri u mreži, gdje se nalaze, koliko ih ima i kakve su im konfiguracije.
U analizi mreže potrebno je znati i koji **mrežni servisi** se koriste i na kojim računarima. Kako se pronalaze računari? Ima li u pozadini DNS ili neki od drugih servis imena (DNS, NIS, WINS, ...)? Postoji li **web server** (HTTP)? Gdje su web serveri i ko njima upravlja? Postoji li **mail server** (SMTP, IMAP, POP, ...)? Postoji li **peer to peer** saobraćaj? Da li je on ovlašten ili nije?
Odgovornost: samo jedna osoba je zadužena za obavljanje cijelog posla
Ukoliko ste dio uigranog tima, dosta je lakše obavljati posao, mada to nije čest slučaj.
Ukoliko ste vođa, onda sva odgvornost leži na vama.
Potrebno je znati koje odgvoran za pojedine računare, mrežne elemente, servise?
Odgovore na sva ova pitanja se mogu naći u dokumentaciji prikupljena tokom analize mreže.
SNMP – protokol koji omogućava direktnu mrežnu komunikaciju sa uređajem iz mreže i dobavlja podatke o njegovom hardware profilu. Bez SNMP-a, automatsko prepoznavanje hardware-a bi bilo problematično.
Vizualni (i fizički) pregled umrežavanja je veoma bitan aspekat u analizi mreže.

Nazivi računara

Računar može imati veliki broj imena. **Host ID** ime je vezano za hardware računara.
Hostname je vezan za operativni sistem, odnosno mrežu. **MAC** adresa predstavlja također jednu vrstu imena – adresu sloja veze podataka. **IP adresa** i domensko ime **DNS** također predstavljaju ime računara.

DNS informacije

Nslookup – komanda koji omogućava pravljenje dns upita mimo aplikacija (postoji i na Windowsu i Unixu). On pročita adresu servera iz konfiguracije mrežne kartice.
Mail exchanger - u svakom domenu koji posjeduje mail server, postoji IP adresa koji će primati mail za taj domen. Preko ove komande (ms) možemo saznati datu IP adresu.
Name server – pomoću komande ns možemo saznati sve name servere (i njihove internet adrese) unutar jedne domene. To će nam dati informacije o host-ovima koji se ne nalaze u našoj lokalnoj bazi.
Pomoću komande nslookup moguće je izlistati sve računare na jednom domenu (omogućavanje toga sa sigurnosnog aspekta nije dobra praksa).
Također, nslookup nam omogućava da promijenimo trenuntni server na kome se nalazi naš računar.
Na Linux-u postoji alat koji se zove **dig**, koji u suštini obavlja sličan posao kao nslookup (zapravo je nslookup *depricated* na Linux-u). Ispisuje sadržaj DNS zone file-a.

Povezanost mreže

Pomoću ping komande provjerava se da li je moguće mrežni podatkovni paket poslati do odredišne adrese bez grešaka (provjera konektivnosti IP adrese na **mrežnom sloju**).
Traceroute – ispisuje sav put koji je se pređe do odredišne IP adrese (što uključuje i ostale servere, odnosno rutere).
Unosom komande **ping** na Windwosu, izvršiti će se slanje paketa četiri puta, dok na Linuxu sve dok ne prekoračimo bandwith svoj (ili eventualno servera kojeg ping-amo).

Traceroute je koristan, ukoliko je došlo do određenog kvara na mreži. Možemo vidjet nad kojoj od prikazanih konekcija nećemo dobiti odgovor, tako možemo saznati da li je problem lokalan prilikom povezivanja s provider-om (ili provider ne dobija odgovor od svog nadprovider-a itd.).

Analiza mrežnih servisa

Kroz dokumentaciju (ukoliko ista postoji) potrebno je otrkiti i locirati postojeće mrežne servise.

Nmap – predstavlja najpopularniji mrežni skener. To je zapravo komad software-a koji otkriva šta to sve postoji na mreži. U principu on pošalje ping na sve IP adrese u mreži i ukoliko mu se neko javi, on zaključuje da na toj lokaciji postoje računari i servisi, odnosno vrši se mapiranje istih.

Uspostavljanje mrežnih servisa

Prvo je potrebno odlučiti se na kojem računaru, odnosno odgovarjućem hardware-u želite instalirati servis.

Potrebno je odrediti koji računar treba da pruža koji servis (zavisno od hardware-a računara i potreba korinsika).

Važan korak pri uspostavljanju mrežnih servisa jeste i organizacija podjele diskova na mreži, tj. potrebno je odlučiti gdje će se ti diskovi nalaziti unutar mreže.

Potrebno je provjeriti da li je potrebno podijeliti zadatke među mašinama i kako sinhronizovati njihov rad.

Odlučiti kakvi će se diskovi koristiti na računarima i da li će to biti fizički diskovi. Također, treba provjeriti da li na postojećim računarima ima dovoljno prostora ili je potrebno nabaviti dodatnog.

Da li ćete iskoristiti više RAM-a? Da li ćete ugraditi brže hard disk-ove u računar, kao i brži NIC (vezu)?

Da li će se na jednom fizičkom serveru nalaziti jedan mrežni servis ili će postojati jedan „jaki“ računar na kojem će se nalaziti svi mrežni servisi.

Redudantnost – da li možemo prihvatiti da servis ne radi u određenom periodu, ili je potrebna njegova konstantna dostupnost.

Servise je potrebno prilagoditi i učiniti prilagodljivim.

URL – usvojen način imenovanja i identifikovanja resursa (uglavnom na web-u)

Funkcionalno imenovanje resursa – imenovanje resursa mora imati veze sa funkcijom koju obavlja (Mail server nećemo zvati DNS , jer to nije njegova funkcionalnost).

Hijerarhijsko ime resursa - /lokacija/računar/sadržaj - dosta organizovan način imenovanja resursa

Jedinstvenost imena - ne bi trebala postojati dva različita resursa sa istim imenom (Unikatni objekti trebaju imati unikatna imena)

Alias – dodjeljivanje nadimaka određenim objektima

Prilikom odabira računara za određeni servis, pored efikasnosti istog u obavljanju te funkcije i sigurnosti), bitna je i mrežna lokacija. Servis koji pruža usluge samo unutrašnjim korisnicima, treba biti u unutrašnjoj mreži i ne smije biti dostupan vanjskim korisnicima.

Međuzavisnost servisa – DNS je jedan takav servis, koji ukoliko ne radi, automatski prekida rad i ostalih povezanih servisa (ukoliko ne radi DNS na etf.unsa.ba, neće ni Zamger, niti Webmail).

Predavanje 6 - LDAP i AD

Pitanje 1: Prava i obaveze korisnika sistema definisane su korisničkom politikom

Pitanje 2: Interaktivni korisnici su korisnici pod čijom se prijavom izvršavaju aplikacije (?)

Pitanje 3: U operativnim sistemima lozinke se čuvaju heširane (kod kriptovanja se očuva povjerljivost, treba ključ; kod heširanja se ne može dobiti početna informacija). Niko ne može znati te lozinke.

Pitanje 4: Grupe nam trebaju radi lakše kontrole prava pristupa. U jednom trenutku grupi damo određena prava, svi korisnici u grupi imaju ista prava. Prava dodjeljujemo na nivou grupa, a u grupu dodajemo korisnike.

Pitanje 5: Passwd - podaci o korisnicima, tu se ne čuvaju šifre (Linux); SAM (Security Account Management) - Windows OS.

Pitanje 6: Passwd i Shadow - razlika je što u passwd se nalaze korisnički podaci osim šifri, a u shadow šifre; veza između njih je na osnovu username-a.

Pitanje 7: Neinteraktivni korisnici (?) Interactive korisnik - prijavljen na sistem i koristi ga

Pitanje 8: Jedan korisnik smije koristiti jedan račun. Razlog: zbog odgovornosti.

Pitanje 9: Kvota - iznos prostora koji se dodijeli jednom korisniku (?)

Pitanje 10: Politika: Napraviti upozorenje koje korisniku daje do znanja da dolazi do kraja svoje kvote, ako on tada ne počne brisati datoteke radi oslobađanja datoteke, onda bi stupilo brisanje datoteka (ili najstarijih ili najmanje korištenih). Korisnici znaju politiku!!!

Pitanje 11: Nije dobro se prijavljivati kao privilegovani korisnik jer sve što uradimo ima maksimalne privilegije.

Pitanje 12: Dodavanje korisnika kroz komanda ili kroz GUI - prednost dodavanja kroz komande (u skriptu) doda se više korisnika automatski

Pitanje 13: Koliko grupa može 1 korisnik biti član - može i smije koliko god hoće (?)
LDAP i AD:
-Termin directory može se prevesti kao imenik. To je baza podataka koja je prije svega namijenjena za pretraživanje. Za razliku od tipičnih baza podataka, imenik se puno rjeđe ažurira (npr. tel imenik se ažurira jednom godišnje).
-Imenici mogu biti **lokalni** (odnosi se npr. na neku organizaciju) i **globalni** (svjetski imenik, npr. BiH - BA; DNS)
-Mogu biti **centralizirani** (nema jedan; primjer je diskusija o seljenju mailova na cloud. Ideja je bila da svi studenti na univerzitetu imaju univerzitetski mail. Pitanje je da li će se podaci držati u UTIC-u ili na cloudu ili će svaki fakultet to organizirati; gdje je originalni, te kako se podaci razdjeljuju) ili **distribuirani** .
-Imenici imaju namespace, koji je vrsta stabla. U imenu se čuvaju opisne informacije, tj. vrijednosti nekih atributa (primjer: atribut je broj telefona, vrijednost je onaj broj koji piše).
-Koristi se i termin repozitorij tj. kaže se da se ovakvi podaci čuvaju u repozitoriju.
-Na početku Interneta, yahoo je bio dominantan pretraživač i bio je imenik. Pretraga se sastojala od toga da kad unesemo podatak, on na osnovu spidera(crawlera?) pretražuje u lokalnoj bazi podataka pojmove koji su vezani za to. Ideja je bila da ljudi svrstavaju web stranice u neke kategorije (npr. fudbal, mediji i sl). Prilikom pretrage se prvo se prođe kroz hijerarhiju. Time se omogućavalo da naša pretraga bude vezana za site-ove koji pripadaju određenoj kategoriji. Kako je internet porastao, ovo je postalo neodrživo.
-Imenici pružaju usluge pregleda i pretraživanja, te ažuriranja. Npr. imenik na telefonima omogućava da ne moramo pamtiti brojeve, nego pamtimo imena, nadimke. DNS je primjer pretraživanja po domenskom imenu. Ovo olakšava administraciju, te omogućava integraciju i centralizaciju.
-Imenik je ustvari baza podataka. Možemo uzeti oracle, sa 1 ili 2 tabele. Međutim, softver koji instaliramo je komplikovan, skup i troši više resursa nego što je potrebno.
-Imenici su optimizirani za ono što je potrebno, a to je uglavnom čitanje. Jednostavnije su informacije, statične su, nema puno relacija. Softver koji se za ovo koristi je mnogo jednostavniji i drugačiji.
-Centralizacija - ako imamo aplikaciju koja je web server gdje se korisnici prijave sa username. Ona ima svoju bazu. Imamo email aplikaciju, gdje se korisnici prijavljuju sa svojim emailom, tj. korisničkim podacima. Svako od njih može imati svoj imenik, tj. informacije o korisnicima, što je komplikovano jer šta se desi ako neki korisnik promijeni lozinku u jednom imeniku, šta se dešava u drugom? U tom slučaju imamo višak informacija. Puno je jednostavnije imati jedan centralni, zajednički imenik, kakav postoji na našem fakultetu. Na ETF-u možemo kroz Zimbru da održavamo LDAP imenik, da ažuriramo naše podatke. Taj imenik koristi c2, c9, zamger.
-Ako imamo iste informacije, treba da se čuva jedna njihova kopija, osim za sigurnosne potrebe, tj. da se ne bi podaci izgubili.
-Prvi imenici su bili zasnovani na x500 standardu.
-ITU(međunarodna telekomunikaciona unija) je napravio protokol, a to je LDAP, koji je danas dominantan protokol. U skladu je sa RFC. Trenutno je v3.
-Kod LDAP-a se imenik definiše kao kolekcija objekata. Kompletan imenik je definisan na osnovu DIT (Directory Information Tree). Pojedini unosi (listovi stabla) su definisani sa Distinguished Name. Svaki DN je grana na stablu.
-Hijerarhija - kreće od root-a. Posljednja komponenta je na najvišem nivou.
-Komponente mogu imati svoja proizvoljna imena, tj. možemo napravit svoj LDAP direktorij, koji koristiti svoja imena. Uobičajeno je da se koriste standardna imena, c - naziv države, o-organizacija itd.
-Relative DN su dijelovi ovoga (distinguished - razlučivo).
-Ime je relativno razlučivo - to znači da na ovom nivou grane ne može biti neko sa istim. Razlučivost na nivou grane.
-DN je zapravo ključ po kojem se vrši pretraga. Mi ustvari tražimo za određeni DN, koji na jedinstven način imenuje jedan objekat u direktoriju, na osnovu tog imena, mi odlazimo do atributa. DN služi da na jedinstven način pročitamo jedan unos u bazi.
-DIT navodi sve grane, tj kompletnu hijerarhiju kao objekte i njihove izglede tj. klase objekata. Tj. kako izgleda grana i koji su elementi listovi - šema baze podataka. Svi unosi koji se nalaze u bazi, ali ne sadržaj nego forma.
-Konkretan unos (list u stablu) je objekat. Klasa definiše kako izgleda objekat i šta je njegova namjena. Definiše koje atribute taj objekt ima. Tu se definiše koji su atributi obavezni, koji nisu.
-Najčešći atribut je objectClass. Svi atributi koji su vezani za neku klasu i koje vrijednosti mogu imati.
-Primjer objectClass
-Sa stablom se definiše hijerarhija, a sa objektima definiše se gdje se šta pohranjuje, tj. kako izgleda nešto što je pohranjeno u imeniku.
-Izuzetak u odnosu na baze - neki atribut može imati različite vrijednosti, primjer: mailovi

(mail i alternativa, ali oba ukazuju na istu osobu)
-Atribut - ima svoj naziv, opis (šta je namjena), tip,, kakve vrijednosti, sintaksa(jel npr. Telefonski broj, ako je mail mora imati @), duzina i OID(neki jedinstveni broj koji definiše objekat). Unutar nekog unosa postoji niz atributa. Atributi su neki predefinisani tipovi, npr. Binarni atribut, string (case sensitive ili insensitive), telefon, DN, vrijeme,...
-Za svaki od unosa mi kažemo kojeg je tipa.
-Kad hoćemo da eksportujemo podatke iz imenika, koristi se LDIF. On se sastoji od toga da se svaki objekat iz hijerarhije ispiše sa svojim DN.
-Za šta se koristi - AD je zasnovan na LDAP-u (28. Slide, predavanje)
-LDAP- protokol koji omogućava komunikaciju sa direktorijem i dodavanje, brisanje i izmjenu unosa, primjenu.
-Prije pristupa imeniku, korisnik mora da potvrdi identitet. Imenik može biti **javan** , pri čemu ne treba potvrđivati identitet za, recimo, čitanje, ali za pisanje mora. Zavisno od toga, LDAP podržava da možemo imati korisničko ime i lozinku. Može se koristiti Kerberos. (provjeriti)
-Sam protokol je klijent server protokol.
Klijent pristupa preko uobičajenog porta 389 i uspostavlja sesiju, povezuje se sa imenikom, unosi korisničke podatke, vrši pretragu, dodavanje, promjene.
-Prvi zahtjev za povezivanje - request. Tu se definiše naziv imenika, verzija LDAP-a te način utvrđivanja identiteta. Kad se poveže, može vršiti pretragu koja obično ima **base** (grana ili jedan unos u stablo), **scope** (obim pretrage - pretraži samo ono na određenom nivou hijerarhije, znači samo grane ispod tog ili kompletno podstablo), **veličinu** , **vrijeme** (“vrati mi ono što možeš u određenom broju sekundi”), **attributes** , **attrsonly** (ili da vraća samo atribute ili atribute i vrijednosti), **filteri** (određuju se neki kriteriji, npr. Tražimo gdje je objekat tipa student) (filteri su 34. slide na predavanju)
-Pretraga vraća i neki kod (povratna vrijednost). LDAP ima dobru šemu, da se na osnovu povratnih vrijednosti zaključi kakav je bio rezultat i šta se desilo
-Za pristup direktorijima se koristi uglavnom API koji ima predefinisane funkcije. Najpoznatiji su Netscape i RFC
-Generalno, pristup LDAP-u se svodi na pozivanje nekoliko fja. Za otvaranje konekcije se poziva funkcija ldap_bind (38 slide predavanje), povratna vrijednost ili success ili broj greske.
-bindova postoji vise, najjednostavniji je ldap_simple_bind(39. slide)
-primjer upita ldap_search_s (40. slide)
-Prolazak kroz rezultate se zasnica da na osnovu pokazivača učitamo prvi i iteriramo (jednostruko povezana lista), prolazak kroz atribute se radi slično
-Bitno je planiranje imenika. Lako je napraviti imenik ako znamo za šta služi. Prvo treba odrediti šta će se nalaziti u imeniku. Zatim, treba odrediti izvor podataka i kakvi su ti podaci (ako kažemo predmet, tj spisak predmeta koji ulaze u imenik, treba odrediti u kakvom obliku su zadani ti predmeti; da li je u pitanju naziv predmeta, šifra predmeta, broj studenata i sl), hijerarhija(ko unosi podatke itd.), da li će biti centralizovan ili distribuiran, kako će se osigurati, skalabilnost, topologija?
-Da bi se Single sign on napravio, treba imati imenik svih korisnika
-Microsoftova izvedba je Active Directory. Tamo se sve pohranjuje, korisnici podešavanja itd. Microsoft je napravio dobro provođenje politike, možemo definisati šta koji korisnik može na kojoj lokaciji.
-Proces prijavljivanja na domen ide kroz ldap, ima svoje resurse, od 2008. se zove Active Directory Domain Services
-Ideja AD je bila da podržava otvorene standarde, da omoguće skalabilnost (da se jednostavno dodaju nova imena), jednostavna administracija(da je sve na centralnom mjestu i da imamo pristup svim resursima) i kompatibilnost (savremeni AD nije kompatibilan sa prvim izvedbama) (provjeriti)
-LDAP ima API niskog nivoa za AD (bitno za projekat jer ćemo koristeći API prilikom prijavljivanja korisnika na mail provjeravati username i password)
-Implementiran je x500 i hijerarhijska struktura
-DNS je neophodan dio AD
-Kerberos se koristi za potvrđivanje identiteta
-AD se može sastojati od više domena (domen je osnovni objekat AD) i dovoljno je da postoji 1 domen da bi već imali active directory. Domena može biti više i mogu biti organizovane u različite strukture.
-Šta se u AD pohranjuje - korisničke grupe i računari (prve izvedbe). Novi elementi su distribution lists, policies ili bilo šta što nam padne na pamet. Svaka aplikacija može definisati svoj objekat, tj. Možemo da dodamo granu hijerarhije, objekt klasu, objekt klasi ću dodati atribute itd.
-Šema opisuje hijerarhiju, sve vrste objekata pored onih standardnih, koje atribute imaju, koji su tipovi.
-Unutar cijelog AD (forest) je jedna konzistentna šema. Unutar jednog skupa domena koje pokriva AD koristi se jedna jedinstvena šema. Šema može biti proširiva, možemo dodati

nove objekte.
-Domen je osnovni gradivni blok, on se nalazi na kontroleru domena (tj. Na računaru koji predstavlja kontroler domena). Unutar jednog domena se radi replikacija svog saobraćaja, sistemske politike i administracija. Domen je nezavisna jedinka - jedan administrator, jedna administratorska politika i sve što se primijenjuje, primijenjuje se na nivou jednog domena.Šema je za cijeli AD (pr. Domen - fakulteti, a šema je na nivou univerziteta).
-Unutar domena mogu biti organizacione jedinice. Ako je ETF domen, odsjeci su organizacione jedinice. Administrativni razlozi - lakše upravljati.
-Domeni mogu biti organizovani u stabla, gdje zapravo imamo domen i neke poddomene (etf.ba je domen, poddomene su odsjeci). Stablo se kreira na način da se na jedan domen dodaju poddomeni.
-Druga organizacija domena je šuma(forest), kombinacija stabala. Na istom nivou hijerarhije doda se još jedan domen i tako se formira šuma. Domeni su odvojeni hijerarhijski, nema nasljeđivanja, svi su ravnopravni na jednom nivou. Prenos povjerenja se podrazumijeva. Ako je korisnikov identitet potvrđen unutar domena, stabla ili šume, on ima pravo pristupa uslugama u drugom domenu.
-Domeni imaju način modeliranja fizičke strukture. Organizacija može imati organizacionu hijerarhiju, a može imati i fizičku hijerarhiju.
-Location site definiše fizičku insfrastrukturu. Predstavlja nešto što se nalazi na jednom mjestu. Ona može pridruživati(?) više domena. Također, jedan domen može biti distribuiran na više lokacija. (Primjeri?)
-Server unutar AD može biti član AD ili domen kontroler. Domen kontroleri - 66.slide
-Kerberos se koristi da bi se prilikom prijavljivanja - utvrđivanja identiteta, utvrđivanje prava i korištenje usluga
-Pošto je u domen kontroler pohranjeno sve što se tiče active directory-a, treba uvijek imati bar dva domen kontrolera.
-Lociranje bilo čega unutar domena se radi na osnovu DNS-a.
-Kerberos - standard;
-Prava za korisnike i grupe se dodjeljuju na nivou grupe. Pri čemu grupe mogu biti lokalne i globalne. Lokalne grupe su grupe vezane za(?), a globalne grupe su grupe koje su vezane za (?)
-Da bi korisnik iz globalne grupe dobio pravo na lokalnom računaru, on mora postati član lokalne grupe. Globalne grupe postaju članovi lokalnih grupa.
-AD od 2008 - s jedne strane korisnički accounti su samo jedan dio toga, a čuvaju se i druge stvari, informacije da li je šuma, koji su domeni ispod šuma, koji se server nalazi u domenu itd.
-Način pohranjivanja informacija o korisnicima i o raznim vrstama podešavanja danas se radi kroz LDAP. To je relativno jednostavan protokol koji omogućava pravljenje hijerarhijsku organizaciju koristeći objekte i standardne nazive grana. Active directory je jedna izvedba LDAP-a.

Predavanje 7 - Domain Name System (DNS)

Odgovori na pitanja:

Razlika između stabla i šume - šuma se sastoji od stabala.
Veza između DN i RDN - DN je jedinstveno ime koje identificira neki objekat. Sastoji se od RDN.
Struktura baze je definisana u shemi.
LDAP provjerava identitet korisnika tako što poredi uneseno korisničko ime i lozinku sa onim koji su spašeni u bazi podataka.
Lokalne grupe su definisane na računaru dok su globalne definisane unutar AD-a. Domenski (globalni) korisnici dobijaju prava na računaru tako što se dodaju u lokalne grupe. Modeliranje logičke i fizičke strukture organizacije - Putem domena i organizacionih jedinica se modelira logička struktura, dok se fizička struktura odnosi na lokaciju.
Koje informacije je potrebno navesti prilikom korištenja LDAP-a kroz neku aplikaciju? – IP adresa računara na kojem se nalazi imenik, broj porta i eventualno verzija LDAP-a. LDAP Interchange Format je standardni format za razmjenu podataka vezanih za LDAP. AD je Windows-ova izvedba LDAP-a.
DN je adresa objekta, a klasa predstavlja strukturu objekta.
Potencijalno pitanje ove godine: Prikaz nekog upita i pitanje šta on znači ili šta neki njegov dio znači?

DNS

Usluge aplikativnog nivoa

Mrežne usluge koristimo isključivo kroz aplikacije, tj usluge aplikativnog nivoa.
Bilo kojoj usluzi koja se nudi preko mreže se pristupa uz pomoc IP adrese kojom se identificira računar na mreži i TCP/UDP porta kojim se identificira aplikacija na tom računaru.

DNS - Domain Name System

DNS je usluga koja omogućava mapiranje između domenskih imena čvorova u mreži i njihovih IP adresa. Nije prava (korisnička) aplikacija, tj ne koriste ga ljudi (direktno) nego aplikacije (koje koriste ljudi) Predstavlja neki vid distribuirane baze podataka. Cilj DNS-a je imati imenik koji se može jednostavno ažurirati i brzo pretraživati.

Aplikacije. kao što su web browseri, kojima trebaju usluge DNS-a se obraćaju servisu koji se naziva *dnsresolver* koji za njih obavlja razrješavanje domenskih imena u IP adrese.

DNS elementi:

● Prostor domenskih imena
● Baza DNS podataka
● Poslužitelji imena (Name Servers)
● Programi za upite – DNS klijenti (Resolvers)

Prostor domenskih imena

Prostor domenskih imena predstavlja skup svih mogućih domenskih imena. Domen u ovom kontekstu predstavlja grupu računara pod kontrolom jedne organizacije Organizacija domena je hijerarhijska u vidu stabla.
Domen može imati poddomene. Cjelokupni Internet je podijeljen manji broj osnovnih (top) domena, a svi ostali domeni su poddomeni osnovnih
DNS osnovni domeni (stari):

● **GTLD** (Generic Top Level Domains)
– .com
– .edu
– .gov
– .mil
– .net
– .org
– .arpa - služi za testiranje
● **ccTLD** - dvoslovni kod države
– .ba (naš, a ima ih preko 250 hiljada milijardi)

Internet Corporation for Assigned Names and Numbers (ICANN - #AJKAN) - na nivou svijeta upravljaju dodjelom domenskih imena i IP adresa. Oni se brinu za podjelu svih IP adresa na 5 registara za 5 kontinenata. Takođe, oni određuju top level domene i upravljaju njima.
Domene se ne prodaju nego izdaju, tako da ETF nije vlasnik domene etf,unsa.ba nego samo korisnik ove domene.

Baza DNS podataka

Ova baza sadrži zapise o resursima (eng. *Resource Records* - **RR**).
Upit ka DNS serveru vraća zapise (RR) vezane uz to ime, pri čemu jedan takav zapis ima sljedeći format:

- Domain_name – ime po kom se pretražuje
– **TTL** – u sekundama – označava koliko dugo je odgovor validan. Na primjer, ukoliko je TTL 3600 to znači da u narednih 60 minuta možemo koristiti podatke koje smo dobili u ovom RR, tj. možemo ih spremiti u keš. Nakon što prođe 60 minuta, nemamo garanciju da je ovaj zapis validan i moramo praviti novi DNS upit.
– **Class** – IN (za Internet), može i drugo
– **Type** – SOA, A, MX, NS, CNAME, PTR, HINFO, TXT
– **Value** – vrijednost, zavisno od tipa zapisa (npr. ukoliko je tip zapisa **A** onda ce ovo polje sadrzavati IP adresu)

Kako radi DNS?

Proces razriješavanja domenskih imena ima sljedeći tok:
-Korisnički program traži IP adresu za ime nekog domena (npr. prilikom pristupa nekoj web stranici).
- Resolver (u lokalnom računaru) pravi upit za definisani name server (najčešće je to lokalni name server koji je u istom domenu).
-(Lokalni) name server provjerava lokalnu bazu:
– Ako pronađe, vraća IP adresu onom ko je tražio
– Ako ne, pita druge dostupne name server-e, pri čemu počinje od korijena DNS stabla
ili koliko visoko na stablu je moguće
-Korisnički program dobiva traženu IP adresu ili poruku o grešci.

Radi uštede vremena i mrežnih resursa dobiveni odgovori na DNS upite se pamte na određeno vrijeme (**cache**) na više mjesta u lancu upita
– Aplikacije (Web preglednici)
– OS
– (Svi) Name server-i koji su se našli na putu upita

Odgovor se pamti onoliko dugo koliko kaže TTL u odgovoru. Svaki name server ima svoju lokalnu bazu podataka i svoj keš. U ovom kešu se nalaze informacije o onim domenskim imenima koja se NE NALAZE u lokalnoj bazi podataka.
Razmotrimo šta se dešava kada name server dobije upit na koji nema odgovor niti u lokalnoj bazi niti u kešu. U ovoj situaciji postoje dvije vrste ponašanja:
- **Rekurzivno** - Obraća se drugom name serveru i vraća dobijeni odgovor. Ukoliko se želi ovakvo ponašanje onda upit koji se šalje mora imati postavljeno polje RD (*Recursion Desired*). U tom slučaju ce biti korišteno rekurzivno ponašanje ali samo ukoliko je to moguće. Na sljedećoj slici je prikazano kako se odvija komunikacija na ovaj način.
- **Iterativno** - Vraća IP adresu sljedećeg name servera kome treba biti upućen upit. Na

sljedećoj slici je prikazano kako se odvija komunikacija na ovaj način.
Ukoliko je upit došao od DNS klijenta (resolver) onda se koristi rekurzivno ponašanje, dok DNS serveri međusobno koriste oba pristupa u zavisnosti od upita i mogućnosti pri čemu se više koristi iterativni pristup. Razlog je bolja raspodjela poslova. Naime, pretraga počinje od root name servera te ukoliko bi se stalno koristilo rekurzivno ponašanje onda bi ovi serveri postali preopterećeni.
Najčešće name server ima zapisane informacije o svim name serverima koji se u hijerarhiji nalaze neposredno ispod njega. Na primjer, root name server ima informacije o .com, .net, .org (i tako dalje) name serverima.

DNS hijerarhija

Prilikom razriješavanja domenskih imena cilj je doći do autoritativnih name servera. Na primjer, prilikom razriješavanja domenskog imena webmail.etf.unsa.ba cilj je doći do name servera od etf.unsa.ba koji u svojim *zone* fajlovima ima zapisane adrese od webmaila, c2, zamgera itd. Neautoritativan odgovor je onaj koji je došao iz keša. Kompletan internet domen je podijeljen na zone. Zone predstavljaju dijelove DNS stabla koji se ne preklapaju, pri čemu svaka zona ima svoj name server. Zone mogu imati proizvoljan broj hijerarhijskih nivoa. Na sljedećoj slici su zone obilježene plavim pravougaonicima.

Root name servers

Postoji 13 root name servera po cijelom svijetu (a-m.root-servers.net). Ovi serveri su fizički rasprostranjeni na 229 lokacija i svaki od njih sadrži iste zapise. Razlog za ovakvu infrastrukturu jeste veća pouzdanost kao i veća brzina odgovaranja na upite. Svaki name server (na svijetu), tj. svaki softver name servera sadrži u konfiguracijskim fajlovima adrese ovih 13 root name servera.
ANYCAST tip slanja - omogućava slanje paketa u slučaju kada se odredišni domen (jedna IP adresa) nalazi na više fizičkih lokacija (čvorova), pri čemu će paket biti dostavljen na najbliži čvor.
Kada u **nslookup** konzoli ukucamo neko domensko ime, npr etf.unsa.ba, dobicemo odgovor u kojem se navodi adresa za etf.unsa.ba ili cemo dobiti informacije o tome kome da se sljedećem obratimo (iterativni pristup). Informacije o serverima kojima se sljedećim trebamo obratiti su izlistane u **Served By** sekciji.

DNS server softveri

● **BIND -** prva UNIX implementacija
- najčešće se koristi - 10 root name servera koristi BIND
● **djbdns** (TinyDNS) – sigurnost prije svega
● **Microsoft DNS** (2000, 2003, 2008, NT4)
● **NSD** – 3 root name servera

BIND - Berkeley Internet Name Domain

Postoji master i slave server, pri čemu na se na master serveru nalaze svi podaci koji se kopiraju na slave servere. Slave serveri služe da bi se rasporedilo opterećenje i povećala pouzdanost.
Najčešće se instalira u poseban direktorij koji se naziva named ili dns koji sadrži poddirektorij master (i slave). U master direktoriju se nalazi datoteka koja ima isti naziv kao i domen, npr. etf.unsa.ba. Takođe, unutar ovog direktorija se nalaze i datoteke koje služe za obratni upit (upit kojim se trazi domensko ime na osnovu IP adrese). Datoteka named.cache sadrži IP adrese root name servera dok datoteka named.conf predstavlja konfiguracijsku datoteku koja "objašnjava gdje se šta nalazi" na serveru. Na primjer, na sljedećoj slici se vidi da je zona " **.** " (root) opisana u datoteci named.cache. Na sljedećoj slici se vidi da se informacije o zoni etf.unsa.ba nalaze u datoteci etf.unsa.ba koja se nalazi u direktoriju master. Razmotrimo šta se nalazi u spomenutoj datoteci etf.unsa.ba koja treba da sadrži sve informacije za zonu etf.unsa.ba. Ovo je obična tekstualna datoteka čije ćemo dijelove pregledati kroz nekoliko slika. Ovi dijelovi ustvari predstavljaju ranije spomenute RR (Resource Records). SOA - Start of Authority - sadrži osnovne podatke i parametre Zapisi o name serveru (ns) i mail exchanger-u (mx). Vidimo da je name server **ns** .etf.unsa.ba a mail exchanger **igman** .etf.unsa.ba. Broj koji se pojavljuje u zapisu za mail exchanger predstavlja prioritet.
Zapisi tipa A predstavljaju preslikavanje domenskih imena u IP adrese. Sa slike vidimo da će se za domensko ime majevica.etf.unsa.ba vratiti IP adresa 80.65.65.73. Pitate se šta je for gods sake majevica?? Hint: OOI majevica. Zadaća majevica…
Stani, želiš da kažes da je majevica ustvari c2? Zapisi tipa CNAME predstavljaju zamjenska imene. Na primjer, c2 je zamjensko ime za majevicu. Dodavanje novog poddomena (umjesto upitnika broj ofc):

DNS pitanja i odgovori:

1. Ako je na vašem računaru kao DNS server podešen server etf.unsa.ba domene, objasniti na koji način će vaš računar doći do IP adrese čije (izmišljeno) domensko ime je www.etf.edu.cn? (Pretpostaviti da za svaki nivo postoji poseban server imena)

Ukoliko pristupamo domenskom imenu preko web preglednika onda se prvo provjerava da li postoji odgovarajući zapis u kešu web preglednika. Ukoliko ne postoji, aplikacija šalje zahtjev lokalnom DNS resolveru na računaru. Resolver takođe provjerava svoj keš i ukoliko ne postoji odgovarajući zapis onda šalje upit lokalnom DNS serveru (ns.etf.unsa.ba). Nadalje pretpostavimo da ni jedan od servera nema zapis u svom kešu. Takođe, pretpostavimo da se radi o iterativnom pristupu. Lokalni DNS server šalje upit nekom od root name servera. Root name server odgovara sa informacijom o adresi name servera za **cn** top domenu. Sada lokalni name server šalje upit name serveru za **cn** domenu i dobija odgovor sa IP adresom name servera za **edu.cn** domenu. Napokon, lokalni name server šalje upit name serveru za domenu **edu.cn** te dobija odgovor sa IP adresom za traženo domensko ime **etf.edu.cn** . Ovaj odgovor se proslijedi lokalnom DNS resolveru koji zatim isti proslijedi aplikaciji koja je prvobitno zahtijevala rezrješavanje domenskog imena.

3. Koja je osnovna razlika između statičkog i dinamičkog DNS-a, i navesti najčešći primjer njihove utpotrebe?

4. Šta **je neophodno podesiti prilikom instalacije DNS servera?**

5. Gdje je definisano koliko drugo treba pamtiti (*cache*) DNS odgovore?

DNS odgovori predstavljaju RR (Resource Records) koji sadrže polje **TTL (Time to live).** Ovo polje govori koliko dugo (u sekundama) treba pamtiti DNS odgovor.

6. Na kojoj lokaciji i u kojoj datoteci BIND očekuje da pronađe informacije o preslikavanju domenskih imena u IP adrese za domen (zonu) za koji je nadležan?
Datoteka se nalazi u direktoriju master i naziva se isto kao i domen (zona). Unutar ove datoteke postoje zapisi tipa **A** (Resource Records) koji predstavljaju informacije o preslikavanju domenskih imena u IP adrese.
7. **Ako je na vašem računaru kao DNS server podešen server etf.unsa.ba domene, objasniti na koji način će vaš računar doći do IP adrese čije (izmišljeno) domensko ime je nivo3.nivo2.nivo1? (Pretpostaviti da za svaki nivo hijerarhije postoji poseban server imena i da serveri podržavaju rekurzivno ponašanje. Pomenuti i objasniti predmemorisanje (*cache*) bar na jednom nivou-koraku.**
U pitanju br. 1 je objašnjen pristup sa iterativnim ponašanjem. Ovdje će biti opisano rekurzivno ponašanje,Ukoliko pristupamo domenskom imenu preko web preglednika onda se prvo provjerava da li postoji odgovarajući zapis u kešu web preglednika. Ukoliko ne postoji, aplikacija šalje zahtjev lokalnm DNS resolveru na računaru. Resolver takođe provjerava svoj keš i ukoliko ne postoji odgovarajući zapis onda šalje upit lokalnom DNS serveru (ns.etf.unsa.ba). Razmotrimo sada keš name servera (u nastavku skraćeno NS) ns.etf.unsa.ba. Svaki NS ima lokalnu bazu podataka (*zone* fajlove) koja sadrži zapise za zonu za koju je zadužen ovaj NS. Pored ove baze podatka, svaki NS ima i svoj keš u kojem se pamte DNS odgovori koji nisu u nadležnosti spomenutog name servera. Ovi odgovori su nekada "prošli kroz" NS u procesu rezrješavanja nekog domenskog imena, na primjer bihamk.ba. Smisao keša jeste da ubrza čitav proces rezrješavanja domenskih imena tako što će umjesto nastavljanja "potrage" za autoritativnim NS-om (u ovom slučaju je to NSr za . **ba** domen) vratiti odgovor iz svog keša. DNS odgovori se u kešu čuvaju onoliko koliko to nalaže polje **TTL (Time to live).**
Nakon što je objašnjeno kako funkcioniše keš name servera, pretpostavimo da traženi zapis ne postoji niti u jednom od keševa. Nastavimo sada sa razrješavanjem domenskog imena **neko3.neko2.neko1** . -Lokalni name server ns.etf.unsa.ba šalje upit nekom od root name servera.
-Root name server utvrđuje da ne posjeduje zapis o domenskom imenu neko3.neko2.neko1 pa šalje upit name serveru top level domene **neko1.** -Name server za domenu **neko1** nema informacije o domenskom imenu neko3.neko2.neko1 ali zna informacije (IP adresu) za NS od domene **neko2.neko1** kojem prosljeđuje upit. -Konačno, NS za **neko2.neko1** ima zapis o domenskom imenu neko3.neko2.neko1 te šalje odgovor NS-u za domenu **neko1** koji ga prosljeđuje **root name serveru** . Root name server ovaj odgovor vraća lokalnom NS **ns.etf.unsa.ba.** NS ns.etf.unsa.ba prosljeđuje odgovor resolveru na lokalnom računaru sa kojeg je inicirano razrješavanje domenskog imena neko3.neko2.neko1. **8. Na koji način serveri imena dolaze do adresa *root* DNS servera? Radi čega su im potrebne te adrese?** Svi name serveri sadrže podatke (IP adrese) o svim root name serverima. Ovi podaci su smješteni u datoteku koja se konfiguriše prilikom instalacije samog softvera za name server.
Ove adrese su potrebne jer razrješavanje domenskih imena počinje upravo slanjem DNS upita nekom od root name servera.
10. Na koji način server imena može napraviti poddomen svom domenu i nadležnost za DNS odgovore iz tog poddomena delegirati drugom serveru imena?
11. Koja je namjena alata *nslookup* ? Od kog servera *nslookup* dobiva informacije? Otkud mu adresa tog servera?
Namjena alata nslookup je razrješavanje domenskih imena. Nslookup inicijalno dobija informacije od defaultnog name servera za mrežu u kojoj se nalazi računar na kojem je pokrenut nslookup. Nslookup može dobiti ovu adresu iz postavki mreže na lokalnom računaru. Ukoliko zelimo da informacije dobijamo od nekog drugog servera onda izvršimokomandu **server <<IP_ADR_SRV>>.**
12. Koliko, maksimalno, upita će DNS server morati napraviti dok ne dobije autoritativan DNS odgovor za adresu www.tamo.negdje.osnovni, ako svi serveri daju iterativne odgovore i svaki server je odgovoran za samo jedan nivo hijerarhije? Objasniti odgovor.
1. Upit prema root name serveru
2. Upit prema name serveru za top level domen osnovni
3. Upit prema name serveru za domen negdje
Dakle, ukupno **3 upita** .
13. Kojim alatom i na koji način se može saznati koji server na nekom domenu je zadužen za razmjenu e-pošte?
Alat **nslookup** spomenut u 11. pitanju se može koristiti da saznamo koji server je zadužen za razmjenu e-pošte..
Prvo je potrebno specificirati da nas zanima server za razmjenu e-pošte. Ovo postižemo naredbom
set q=mx , gdje je mx skračenica za *mail exchanger.* Zatim upisujemo domensko ime (npr. google.com) i pritisnemo *Enter.* Vidjećemo da su izlistane informacije o mail serverima za google.
14. Napisati DNS zapise o resursima (RR) koji će iskazati da je server imena za zonu "imeserver" koji se nalazi na adresi 100.100.100.5, da je server kom treba slati
Nedostaje dio pitanja.
RR: **imeserver A 100.100.100.5**
15. Objasniti kako se na Windows serveru u postavkama DHCP usluge promijeni informacija o IP adresi DNS servera.
#justdoit i guess