

## **PREDAVANJE 1**

### **PREDAVANJE 1.1**

- Kako znati da li je problem sa hardverom ili softverom? Ako se počne podizati operativni sistem, najvjerojatnije je problem u softveru. Ako je u hardveru problem onda bi to vjerovatno spriječilo da se podigne operativni sistem.
- Šta može biti ako nemam veze prema Internetu? Ne radi DNS, ruter nije dobro podešen, kablovi ...
- DNS - Domain Name System - Prevođenje domenskih imena u IP adrese.
- Šta je web server? Server - hrvatski prevod je poslužitelj. Web server poslužuje web stranice. Kad ja ukucam u web pregledniku domensko ime, taj zahtjev ode web serveru i on vrati web stranicu.
- Šta je sigurnost? Postoje pravila koja definišu šta je zabranjeno a šta ne.
- LDAP- Lightweight Directory Access Protocol
- Kad se prijavljujemo na fakultetske usluge, imamo jedno korisničko ime i jednu loziku. Postoji neki imenik (baza podataka) gdje se nalaze podaci o svim korisnicima.
- Koliko ima domenskih imena? Red veličine stotine miliona. S obzirom da je to ogromna baza, upit nad tim bi trajao danima.
- WWW koristi http.

### **PREDAVANJE 1.2**

- Android operativni sistem je usko povezan sa hardverskim komponentama pa ga obično nismo sami instalirali.
- Instalacija Android aplikacija je lakša nego Windows/Linux, bolje je približena svakom korisniku.
- U nekim slučajevima na pametnim uređajima korisničko okruženje može biti ograničeno jer nije namijenjeno za to, pa može biti teže pristupiti mrežnim postavkama.
- DHCP protokol koji automatski daje IP adrese. Kod servera je drugačije jer on ima fiksnu adresu.
- AP lozinka: većina korisnika ne zna konfigurirati AP pa vjerovatno je da će nešto zeznuti
- Da bi se program pretvorio iz izvornog oblika u izvršni: napisali smo neki kod, kompajlira se, linkuje se (povezuje se sa sistemskim komponentama, npr pozovem printf->on pozove usluge operativnog sistema tj. neku biblioteku)
- Linkovanje- omogućavanje da se čitav kod izvrši

## **PREDAVANJE 2**

### **PREDAVANJE 2.1**

- Najveći dio download Internet saobraćaja je video (na globalnom nivou oko 56% - Netflix, razne vrste stream-ova, YouTube)
- Sistem kojim upravljamo - računarska mreža se sastoji od korisničkih uređaja: računari, tableti, mobiteli, pametni frižideri, televizori ... na kojima je instaliran neki softver koji ima neku funkcionalnost
- Ti uređaji se međusobno i sa ostatkom svijeta povezuju putem mreže
- Mreža se sastoji generalno od kablova i zraka - u smislu radiotalasa kojim se povezuju, i uređaja koji omogućuju to povezivanje (switch, router, AP i sl.), i posljednja, najvažnija, komponenta su korisnici jer oni čine taj sistem, utiču na ponašanje sistema (oni nisu pasivni, nego neko ko definitivno utiče)

### Hardver

- Hardver je osjetljiv: ispuštite nešto razbiće se, ako ga pritisnete prečvrsto slomiće se, ako ga dirate vlažnim rukama može ga oštetiti statički
- Zašto treba gasiti uređaje prije nego ih isključimo iz struje, a iz struje izvući kabl kad god radimo bilo šta sa računarom? Iz dva razloga:
  1. Teoretski, hard disk bi se mogao oštetiti, mada su male šanse
  2. Kada se ugasi onda se uredno završi čitav proces, znači operativnom sistemu kažemo: Slušaj prestat će radit, napiši sve što trebaš gdje trebaš da zapišeš, završi sve procese, parkiraj glave diskova (nekad je to bilo važno) i sve to završi
- Ne može računar da se ugasi normalno (pritisnemo dugme ništa se ne dešava), šta onda (a da ne izvlačimo iz struje)? Potrebno je držati dugme dovoljno dugo (30 sekundi će sigurno biti dovoljno, 5-10 sekundi je uglavnom dovoljno)
- Kad je računar uključen u struju, šta je uopće pod napajanjem? Kabl dolazi u jedinicu za napajanje i ona u principu osnovnu ploču drži pod napajanjem, tako da

određen napon već postoji.

- Konektori su pravljeni tako da ih ne možemo pogrešno okrenut, međutim ako smo dovoljno uporni i ako dovoljno jako gurnemo, možemo slomit pinove. Često se dešava npr. Na projektoru da nema neke boje ili sl., to je zato što je neko iskrivio kabl
- UPS - ako nestane struje da se nastavi napajanje, služi kao neka vrsta akumulatora. Razlika između UPSova: koliko dugo može da drži sistem upaljen (obično nije potrebno više od pola sata-sat), druga stvar je kad se izgubi napajanje koliko brzo se prebacuje na UPS. Ako imamo sistem od više korisnika, važno je da sistem ne izgubi napajanje.
- Računari su osjetljivi na loše uslove: kad im je vruće, pregriju se i počnu da lošije rade, neki se znaju zbog toga ugasiti, (za hladnoću nije siguran kako im smeta), visoka vlažnost korozija, niska vlažnost statički elektricitet - treba održavati vlažnost (klima za održavanje i vlažnosti i toplote)
- Moramo obraćati pažnju kad kupujemo novi disk da li je on kompatibilan sa našim računarom: da li je njegov konektor i kontroler u skladu sa onim što mi imamo na našem računaru. Interfesji za hard diskove su uglavnom SATA. SATA je serijska konekcija, PATA paralelna, kako je serijski brži od paralelnog kad na paralelnom šaljemo više odjednom ? Svodi se na to da je jednostavniji, u principu dovoljno brzo možemo slat.
- Zašto u računarima imamo i solid state diskove i HDD i kakva je razlika? Nekad je ubrzanje računara podrazumijevalo kupovinu memorije. Prvi SSD su bili flash drive-ovi (malo veći stikovi). Sada za ubrzanje kupujemo SSD. Na SSD stavljamo operativni sistem i aplikacije (250GB dovoljno), na HDD stavimo podatke jer njih ima više (terabajtni dovoljan). Operativni sistem prebacujemo sa jednog diska na drugi tako što prebacimo kompletnu particiju (napraviti backup svega). Danas se i u serverima pojavljuju SSD velikih brzina.
- Memorije treba imati dovoljno. Otovrimo task manager da vidimo koliko je zauzeto. Jedno vrijeme su se memorijski moduli jako brzo mijenjali, pa je trebalo obraćati pažnju prije svega da može stati u slot. Treba da je ista vrsta memorije i da je ista brzina (ako je brzina različita, radiće onom brzinom kojom radi najsporija). Kad se kupuje server treba kupiti što više komponenata i napuniti ga što više (finansijski isplativije).

## PREDAVANJE 2.2

### Operativni sistemi

- Operativni sistemi omogućavaju ?, upravljanje aplikacijama, upravlja datotečnim sistemom (file sistem - omogućava da bite na disku organizujemo u neakve fajlove) i omogućava komunikacija sa korisnicima
- Operativni sistem teoretski može obavljati samo jedan zadatak i opsluživati samo jednog korisnika
- Postoji jedna linija operativnih sistema koji su iznikli iz DOSa i koji su bili jednokorisnički sistemi - bio je jedan korisnik na sistemu, što ima veze s tim na koji način su organizovane neke stvari na sistemu
- Savremeni operativni sistemi svi mogu obavljati više zadataka istovremeno i opsluživati više različitih korisnika istovremeno, potiču od NTa (Windows NT bio prva verzija višekorisničkog OSa)
- Druga klasa operativnih sistema su UNIXoidni operativni sistemi u koje spadaju savremeni Linux, MAC, iOS, Android
- Nema neke posebne razlike sa aspekta administracije, i jedan i drugi OS obavljaju istu funkciju, možda rade drugačije ali danas sve više interfejsa liče jedan na drugi. Ako razumijemo šta radi jedan OS, na koji način radi, šta može uraditi - mi to možemo uraditi i na jednom i na drugom OSu
- Windows je prošao kroz razne generacije: Window10 bi trebao biti posljednji Windows, on bi se trebao ažurirati (kod servera Win10). Ono što je karakteristično je da Windows ima korisničke operativne sisteme (koje su za pojedinačne korisnike) i serverske (koji su u principu pravljeni da opslužuju veći broj korisnika). Linux nema tako jasnu distribuciju, mada ima onih koji su namijenjeni kao desktop i serverska distribucija, razlika je u tome koje su se komponente instalirale prilikom inicijalnog instaliranja operativnog sistema (ako ću ja koristiti svoj računar kao desktop računar, vjerovatno mi na njemu ne treba ni DNS server, ni mail server, ni Web server, itd., ako će to biti server vjerovatno mi ne treba grafičko okruženje). Ideja instalacije serverske ili klijentske verzije OS je da se instaliraju samo one stvari koje su nam potrebne.
- Zašto spominjemo onu varijantu OSa koji je bio namijenjen za samo jednog korisnika? Kućni računar je bio namijenjen za samo jednog korisnika, nije bilo potrebe da odvajamo prava jednog i drugog korisnika, jedan korisnik je istovremeno bio i administrator. Ti računari nisu bili umreženi, sigurnost nije bila pitanje, nije neko mogao sa strane doći na naš računar. Kad su se računari počeli umrežavati, i kad se ispostavilo da jedan računar treba koristiti više ljudi, tada se javila potreba da se odvoje podaci, da se zaštitimo od uticaja izvana, i s druge strane da omogućimo zaštitu između različitih korisnika.

- Misrosoft da ne bi izgubio tržište je morao omogućiti kompatibilnost unazad: instaliramo novi OS, ne bi trebalo da nam stare aplikacije ne rade. Danas Win10 ima jednokorisnički mode (tzv. DOS mode) gdje možemo pokrenuti stvari koje u principu ne bi mogle da se koriste.
- Čim ima više korisnika, potreban je neko ko će određivati koji od korisnika ima na šta pravo. U takvim višekorisničkim sistemima postoji privilegovani korisnik (super user, na Win administrator, na Unix root). Na Win se može promijeniti administrator, npr. Ako neko dođe do njegove lozinke, on se čak može i proglasiti da više nije administrator. Oni imaju manje-više sva prava na sistemu. Služe da pripreme okruženje za rad drugih korisnika, i ne bi trebalo da se računar koristi pod prijavom privilegovanog korisnika, zato što privilegovani korisnik nema ograničenja (napravimo grešku, nešto obrišemo, pokrenemo zlonamjerni softver ... sve što radimo imaće posljedicu na sve korisnike računara). Ako smo prijavljeni kao obični korisnik, sve greške, zlonamjerni softver se uticati samo na nas, kao tog jednog korisnika.
- Kako savremeni Windows osigurava da mi iako se prijavljujemo kao privilegovani korisnik ne uradimo neke gluposti ? Iako smo prijavljeni kao privilegovani korisnik, mi sve ustvari izvršavamo kao obični korisnik. Kad hoćemo da uradimo nešto što zahtjeva administratorske privilegije, u tom trenutku on kaže: Slušaj, ovo što sad hoćeš da uradiš može imat šire posljedice, odnosno hoćeš da uradiš nešto što zahtijeva administratorske privilegije, molim te da potvrdiš. Kad se pojavi ovakva poruka, trebali bi se zamisliti (hoćemo ili nećemo to je druga stvar). Zašto bi se nešto instaliralo kad mi nismo pokrenuli ? Znači mi smo prijavljeni kao privilegovani korisnik, ali kad nešto hoće da se instalira ili obriše, OS nas upozorava da to ne radimo.
- Na Unixoidnim sistemima smo prijavljeni kao grupa korisnika koji mogu preuzeti administratorske privilegije (sudo).
- Kažu da je OpenBSD najsigurniji OS, ali je on siguran onoliko koliko ga mi znamo konfigurisat (ako njega ne znamo konfigurisat, a znamo neki drugi, taj drugi će biti sigurniji).
- Windows nije ništa nesigurniji nego drugi, samo je više izložen.
- Princip minimalnih privilegija - ne bi niko trebao da ima više prava nego što mu je neophodno da obavi svoj posao.
- Operativni sistemi obavljaju istu stvar, samo je način na koji se to radi drugačiji.
- Datotečni sistem omogućava da bite na disku organizujemo u neakve fajlove, da ne predstavljaju niz bita bez značenje, nego neki sadržaj. Način izvedbe nije bitan, bitna nam je funkcionalnost. Bitna su prava pristupa datotekama. Unixoidni datotečni sistemi su uglavnom hijerarhijski organizovani, imamo neki root folder, gdje su određene stvari pohranjene u određene foldere, imaju simbolički linkovi (link koji vodi negdje drugo), i ono što je bitno kod Unixoidnih sistema imaju implementirana prava pristupa, pri čemu postoji ACL skraćenic, gdje imaju 3 vrste korisnika: vlasnik datoteke, grupa vlasnika datoteke i svi ostali, i mogu imati pravo čitanja, pisanja i izvršavanja datoteke (datoteka postaje izvršna tek kad joj dodijelimo prava izvršavanja)
- Windows datotečni sistemi bili su FAT, danas je to NTFS. Svaka particija se prijavljuje kao nezavisni disk od A do C, od A i B su nekad bili za flopi diskove. Ima hijerarhijsku organizaciju, pri čemu je Windows za operativni sistem, Program Files za programe, Documents i Settings (User u novijim verzijama) za instalacije i sl.
- Kod Windowsa ekstenzija datoteke određuje koja je njena vrsta, kod Linuxa ekstenzija datoteke formalno nije bitna, ali danas većina aplikacija čita ekstenziju i na osnovu nje pokreće odgovarajući program da pristupimo datoteci, ali generalno kod Windowsa bukvalno ekstenzija određuje tip datoteke i ništa drugo
- Linux ima pune access control liste (liste kontrole pristupa), znači za svaku datoteku ja mogu da kažem koji tačno korisnik ima pravo pristupa
- Postojali su razni mrežni datotečni sistemi koji su omogućavali razmjenu datoteka između različitih operativnih sistema, međutim danas savremeni datotečni sistemi podržavaju razmjenu datoteka (možemo bez problema napraviti dijeljeni folder na Windowsu, i pristupiti mu sa Linux računara ili obrnuto)
- SMB - protokol koji omogućava pravljenje dijeljenih foldera, podržavaju i Windows i Linux
- Sve što se dešava na OSu, svaki program koji se pokrene, je zapravo proces, pri čemu procesi mogu biti oni koji se izvršavaju u pozadini odnosno procesi koji nemaju direktnu interakciju sa korisnikom (ugl servisi) i procesi koji imaju direktnu interakciju sa korisnikom i oni se izvršavaju pod prijavom korisnika koji ih je pokrenuo i imaju prava korisnika koji ih je pokrenuo. Svaki proces ima svoj ID po kojem se oni prepoznaju. Ako jedan proces pokrene drugi proces, onda je jedan proces roditelj, drugi je dijete. Ako se proces roditelja prestane izvršavati, a proces djeteta ne prestane, onda je to zombi proces i nema načina da se on zaustavi. Windows ima nešto slično. Oba imaju environment variables - to su definisane var kojima svi programi mogu da pristupe i pročitaju -

najčešće za lokaciju da kazemo gdje se nalazi neka datoteka (kad ukucamo path koji folderi će se pretražiti - i to onim redoslijedom kako su upisani)

### RAČUNARSKA MREŽA

- Mreža - međusobno povezana skupina autonomnih računara
- Svaki od računara, svaki od čvorova, svaki od uređaja u mreži je nezavisni uređaj, znači on može raditi sam za sebe, to što je on povezan sa drugim uređajima daje mu ustvari dodatnu funkcionalnost, omogućava mu da razmjenjuje informacije
- ISO/OSI slojevi
- Bitno: fizički sloj - da bi mogli komunicirati, moramo imati neki medij pomoću kojeg ćemo komunicirati. Informacije, signali moraju se prenositi. Dva su načina: Jedan je kroz kablove - kroz neki fizički medij, drugi je bežično, odnosno putem elektro-magnetnih talasa. U suštini uvijek se prenos vrši putem elektro-magnetnih talasa, ali se kod kablova oni kreću u ograničenom okruženju (svjetlost). Prednost kablova: bolje su zaštićeni od vanjskih uticaja, samim tim domet kabla je veći nego što je domet bežičnog prenosa. Prednost bežičnog prenosa: jeftiniji je, jednostavniji je (najčešće), ali je isto tako manje pouzdan.
- Sloj veze podataka - ako smo fizički povezali uređaju moramo na neki način upravljati podacima koji putuju (moramo biti u stanju da kažemo kojem ćemo uređaju na mreži poslati, koja je adresa, da otklonimo greške i sl.). Od raznih protokola danas je dominantan žičani Ethernet protokol. Nekad je bio spor, savremeni Ethernet radi na brzinama većim od 100 gigabita. Druga dominantna tehnologija na data link sloju je bežični WiFi 802.11 protokol (WiFi je alijansa koja potvrđuje da je neki proizvod u skladu sa ovim protokolom).
- Brzina interneta ograničena ISPom, ne wirelessom. Brzina wireless-a je oko 200-300 Mbps.
- Nedavno su preimenovane verzije protokola, sada su numerisani 4,5,6. 6 je najnovija verzija, još uvijek nema rutera koji to rade.
- Ako imamo uređaj, u njemu otvorimo web preglednik - aplikaciju i ukucamo neki url, nešto se desi nekom magijom i onda se unutar preglednika prikaže sadržaj tog url-a. Šta se zapravo desi ?
- Aplikacija pošalje get zahtjev (http request) na ovaj url i od njega dobije http reply i prikaže sadržaj. Taj zahtjev je otišao do web servera. Aplikacija get zahtjev, odnosno poruku proslijeđuje operativnom sistemu, i kaže de mi molim te proslijedi ovo na odgovarajuću adresu.
- U url se nalazi domensko ime
- IP adresa je adresa računara, a port je adresa aplikacije na tom računaru. Znači IP adresa služi da poruka stigne do određnog računara, a port služi da se zna kojoj aplikaciji na tom računaru je namijenjena.
- Operativni sistem uradi dvije stvari: prvo TCP sloj doda TCP zaglavlje (ili UDP) i IP sloj doda svoje zaglavlje. Ono što je nama bitno je da se u ovom zaglavlju nalazi određni port aplikacije. Sad ćemo se spustiti do IPa. I šta je još bitno, IP od transportnog sloja dobije niz bajta, on u svoje zaglavlje između ostalog dodaje IP adresu odredišta. Šta se onda dalje dešava s tim paketom ? Operativni sistem taj paket proslijedi NICu ( Network Interface Card - mrežna kartica), ona dobije IP i ona od tog IPa, prije nego što stavi bite (pošto ona stavlja na žicu niz bita), ona mora dodati data link zaglavlje. Odnosno, ona ovo što dobije od IPa, opet posmatra kao niz bita ili bajta apsolutno nebitnog sadržaja, i na to dodaje svoje zaglavlje (koje, između ostalog, podrazumijeva MAC adresu).
- Ako imamo više PCa utaknutih u jedan switch, kako oni jedan drugom šalju poruke ? Šta radi switch kad dobije paket ? Dolazi paket koji ide na MAC adresu 2. Switch ima tabelu u koju piše MAC adresu i interfejs/port. Switch kad dobije paket pogleda određnu MAC adresu i nađe na koji interfejs treba proslijediti paket, i time on završava svoj posao. To je uredu dok god se ti PC nalaze u istoj mreži (podmreži). Zna li naš switch adresu fejsbuka ? Ne zna, jer nije u toj mreži. Prema tome, očigledno ako šaljemo paket na fejsbuk ne možemo ga ovako poslati.
- Neka je naš računar npr. PC1. On treba poslati paket ne nekom računaru u istoj mreži, nego fejsbuku (čiju IP adresu ne zna). Kako će to ostvariti ? Ovaj PC1 pošalje ARP zahtjev (ARP pretvara neku IP adresu u odgovarajuću MAC adresu) za MAC adresu u kojem napiše IP adresu koja pripada fejsbuku. Taj zahtjev se šalje kao broadcast, što znači da će je switch poslati na sve svoje interfejse.
- U konfiguraciji računara je upisana IP adresa defaultnog gatewaya (to je čvor preko koga možemo pristupiti uređajima koji nisu u našoj lokalnoj mreži).
- Hijerarhija postoji kod DNSa, kod rutiranja nema neke takve vrste hijerarhije. Kod mreža se sve stvari dešavaju lokalno, znači do sljedećeg čvora.
- Očigledno paket za fejsbuk moramo poslati defaultnom gatewayu. Da bi mu mogli nešto

poslat, on mora biti povezan na switch (najčešće je ruter jer on odvaja dvije mreže). Ako switch dobije neki paket koji ide na neki interfejs kojeg nema u tabeli, on ga šalje na sve portove, ili ako dobije broadcast adresu šalje na sve portove. Znači nama treba način da mi poruku za fejsbuk, koja je stigla do switcha, na neki način proslijedimo na ruter. Moramo mu poslati na MAC adresu. Otkud nam njegova MAC adresa

? Preko ARPa: pošalje se svima IP adresa, ruter kaže to je moja adresa. Ta adresa se upiše u tabelu kod switcha, vrati se PCu1 i on onda zna na koju adresu da šalje.

- IP adresa, Defaultni Gateway, DNS, Subnet maska (govori nam fiksiranih bita IP adrese - korisno jer računari na istom subnetu mogu direktno komunicirati, na različitim ne mogu)
- Lokalna mreža: 3 računara, na jednom podignut web server. Kako ćemo poslati poruku od računara PC1 računaru PC3 iz te mreže? PC1 će reći on je u mojoj lokalnoj mreži pa će kao odredišnu IP adresu staviti adresu PC3. Kako on zna da je PC3 u njegovoj lokalnoj mreži? Subnet maska nam služi prije svega da kad PC1 dobije IP adresu njemu je jako važno da li taj paket treba slati na MAC adresu nekog od računara u lokalnoj mreži ili na MAC adresu defaultnog gatewaya, a to zna na osnovu subnet maske. Kako on to zna? IP adresa je 32 bitna vrijednost podijeljena u 4 grupe. Npr adresa se sastoji od: 00000000.00000000.11111111.11111111. (0.0.255.255). Subnet maska je isto 32 bita, pri čemu ima n jedinica i 32 - n nula, npr 11111111.11111111.00000000.00000000. Adresa koja se nalazi u ovoj podmreži (na osnovu ove maske) mora imati prvih 16 bita kao i subnet maska (onoliko koliko je jedinica - jedinice su fiksirani biti). Jedan način da se zapiše ova subnet maska je 255.255.0.0 ili 255.255.0.0/16 - 16 znači da sve IP adrese koje imaju istih 16 bita kao ova u istoj podmreži.
- Dakle osnovna namjena subnet maske je da pomogne PC1. On uzme svoju IP adresu i IP adresu odredišta i poredi ih. Kako neki računar određuje da li je neki računar u njegovoj podmreži? Dva računara su u istoj podmreži ako im je početni broj bita određen subnet maskom isti (odnosno ako im je mrežni dio IP adrese isti).

### PREDAVANJE 2.3

- Računar treba da pošalje paket na IP adresu X, potrebno mu je da zna da li je ta adresa u njegovoj lokalnoj mreži ili van njegove lokalne mreže. To zna tako što uzme svoju IP adresu iz svoje konfiguracije, uporedi je sa IP adresom X na način da uporedi prvih n-bitova određenih subnet maskom. Ako je iz lokalne mreže, kako će onda poslati tom čvoru paket? Ima njegovu IP adresu, zna da je u njegovoj lokalnoj mreži. Znači on će uraditi ARP upit za tu adresu iz lokalne mreže. Taj ARP će dobiti svi, onaj na čiju se adresu odnosi upit će odgovoriti. Kad paket dođe do switcha, on će taj paket proslijediti na odgovarajući interfejs. To što je uređaj dobio paket, ne znači da je taj paket za njega, nego je taj paket možda poslan na broadcast adresu. Znači kad računar dobije paket, pogleda je li njegova MAC adresa i tako zna je li paket za njega, otvori paket i pogleda IP adresu, i ona je njegova. On dalje nastavlja raspakivanje dok ne dođe do porta. Pogleda port (80), proslijedi web serveru i web server dobije taj get zahtjev. Šta ako je IP X adresa nije u njegovoj lokalnoj mreži? Paket treba poslati ruteru. Poslat će zahtjev za adresu rutera, dobiće odgovor i paket će doći i do rutera.
- Kako će ustvari računar PC1 napraviti paket? U njemu će prvo pisati MAC adresu defaultnog gatewaya, onda će pisati IP adresu fejsbuka. To je bitno zato što će ovaj paket doći do switcha, pročitati MAC adresu defaultnog gatewaya i proslijediti na njegov interfejs, i taj će paket doći do defaultnog gatewaya. Defaultni gateway će vidjeti da je MAC adresa njegova, krenuće da ga raspakuje i vidjet da nije za njega nego paket za fejsbuk. Ovako paket dolazi do defaultnog gatewaya i ovako on zna da paket nije za njega.
- Paket koji je dobio defaultni gateway znači sadrži njegovu MAC adresu, IP adresu fejsbuka, port npr 80, get zahtjev. Paket je mogao biti i za defaultni gateway, jer na ruteru može biti npr podignut server.
- U ovom paketu, ruter ne zanima ostatak paketa, zanima ga samo IP adresa.
- Ruter radi dvije stvari: jedna je usmjeravanje/prosljeđivanje - switch kada dobije paket on pogleda MAC adresu i proslijedi ga na odgovarajući interfejs. Slično radi i ruter, on treba da odluči na koji od svojih interfejsa će poslati paket. Ruter radi sa IP adresama. Koristi tabele rutiranja. Kad se i kako formira tabela rutiranja? Ona je prethodno izračunata. Kako on odluči kuda da pošalje paket, odnosno koja ruta mu je najbolja? Rutiranje bi trebalo da mu omogući da zna da dođe do bilo koje adrese. On ima nešto što se zove defaultna ruta: sve što ne zna šta će s njim prosljeđuje na taj interfejs. Teoretski trebalo da ruter zna sve, ali on u praksi zna da dođe do rutera koji su u dijeljenoj mreži. Bitno je da kad paket dođe do rutera, već ima izračunata ruta. Kako se ta ruta izračunava? Pomoću algoritama rutiranja.
- Kad konfigurišemo ruter, svakom interfejsu dodijelimo IP adresu, subnet masku i koji

protokol rutiranja koristi. Ruter kad se upali, pošalje poruke drugim ruterima u njegovoj mreži, oni razmijene informacije ko do koga može doći i nakon te razmjene informacija ruter ima tabelu rutiranja.

- Dakle ruter do kojeg je došla poruka od PC1 namjenjena za fejsbuk je proslijedio treba da proslijedi tu poruku sljedećem ruteru u nizu. Odakle mu adresa tog rutera? U tabeli rutiranja mora postojati nešto što se zove next hop - to je adresa sljedećeg rutera. Saobraćaj koji je išao od PC1 do prvog rutera je isti kao i između ova dva rutera - to je lokalni saobraćaj. Znači prvi ruter će u paket dodati MAC adresu drugog rutera. U drugom ruteru se dešava isto, i tako sve do kraja.
- Fejsbuk je web server. Znači postoji neki računar (sad je to u reku), koji ima IP adresu fejsbuka, koji na nekom portu npr 80 aplikaciju web server i u njoj se izvršava aplikacija fejsbuk.
- Paket je došao do rutera za koji je vezan računar na kome se nalazi server na kojem je fejsbuk (i još računara). Šta će se desiti? Ruter je skinuo zaglavlje, piše IP fejsbuka. Šta će ruter uraditi s tim? Za next hop će pisati direktno povezan - ne dolazim do njega preko drugih rutera, on je povezan direktno na moju mrežu. Kad bude pravio ARP upit, on će biti na IP adresu fejsbuka, vrati mu se MAC adresa, i izlazi paket sa MAC adresom fejsbuka. Kad računar dobije paket, vidjet će da je njegova MAC adresa i njegova IP adresa, a port je 80. Operativni sistem tog računara kaže ima aplikacija koja se prijavila da dobija pakete na portu 80. Toj aplikaciji se pošalje get zahtjev. Aplikacija će dobiti i adresu pošiljaoca, da bi znala da vratiti.
- Za šta nam treba subnet maska? Da pošiljalac odredi da li je odredište u njegovoj lokalnoj mreži ili van lokalne mreže, jer se način slanja razlikuje.
- Za šta nam treba defaultni gateway? IP adresa preko koje, koristeći njegovu MAC adresu šaljemo pakete ka konačnom odredištu.
- DNS pretvara domenska imena u IP adrese.
- Hoće li se paket vratiti istim putem kojim je i otišao? Vjerovatno hoće ali ne mora značiti. To zavisi od tabele rutiranja. Vrlo važno je da taj put neće biti zapamćen: kad ruter dobije paket i proslijedi ga na svoj interfejs i potpuno zaboravi na taj paket.
- CloudFlare - trenutno poznatiji/bolji DNS server (1.1.1.1)
- DHCP dodjeljuje IP adresu, subnet masku, DNS, defaultni gateway.  
Pitanja na ispitu: koja nam 4 parametra trebaju, zašto nam treba svaki od njih?
- Ona kutija koja ima nekoliko različitih funkcija (što imamo kući, ne zna ni on kako se zove) - u njoj se nalazi nekoliko stvari: switch, ima ruter, AP, bridge (uređaj koji sa 802.11 - sa wireless interfejsa pretvara na recimo 802.3), i obično ima i firewall.
- Kad neki PC šalje neku poruku, ona ode do rutera, i ako je na njemu podešen NAT (a na većini jeste), on radi prevođenje adresa. Npr neka PC ima IP adresu A i šalje paket na IP adresu C. Kada taj paket dođe do rutera, i kada ga ruter proslijedi, on dobija IP adresu vanjskog interfejsa rutera npr B. Znači adresa pošiljaoca je sada B, a adresa primaoca je idalje C.
- IP adrese bi trebale da budu privatne. Međutim, ispostavilo se da 4 milijarde adresa koliko ih ima nije dovoljno. Postoji određen skup adresa koje se smatraju privatnim - znači da ruteri na globalnom internetu neće prosljeđivati pakete sa ove adrese, druga, važnija stvar je da mi ovu adresu možemo staviti u svoju mrežu, a neko drugi može u svoju. Kako se osigurava da se neće pomiješati? Upravo zbog NATa.
- Vanjska adresa interfejsa rutera ne mora biti javna, i ona može biti privatna.
- Jedna ideja NATa je da se štede IP adrese, druga je zaštitna zato što svi paketi iz unutrašnje mreže imaju samo IP adresu vanjske mreže, znači niko ne zna unutrašnju strukturu mreže.
- Kad poruka stigne na IP adresu C, odgovor će ići na adresu B, odnosno na adresu rutera. Za koga je odgovor? U ovom slučaju, ruter ne smije zaboraviti paket (kao inače). Onda pravi tabelu u kojoj treba da bude pošiljalac, primalac (odnosno IP adresa A, IP adresa C). Ali šta ako više poruka sa različitih IP adresa iz iste lokalne mreže ide na IP adresu C? Onda se radi nešto što se u mrežama smatra neispravnim: kombinuju se podaci sa drugog zaglavlja i uradi se sljedeća stvar. NAT zapamti IP adresu pošiljaoca A i zapamti port. Onda zamijeni IP adresu A adresom B, ali isto tako zamijeni i port pošiljaoca slučajnim portom. Kako nam ovo pomaže? Npr računar D šalje poruku za port 30 000, i računar A iz iste lokalne mreže šalje poruku za port 30 000. NAT će adresu D zamijeniti svojom IP adresom, port 30 000 će zamijeniti npr portom 10 001, i IP adresu A će zamijeniti svojom adresom, port 30 000 će zamijeniti npr portom 10 002. Kad dobije odgovor, vidjet će da je on za port 10 002 i iz tabele će pročitati da ga treba vratiti računaru A.

## PREDAVANJE 3

### PREDAVANJE 3.1

#### Zadaca

- Kablovski internet ne znači da dolazi kablom. Bežični ide preko telefona. I kod jednih i kod drugih od ISPa ide neki kabl. I jedna i druga imaju splitter, on radi frekventno razdvajanje na dva dijela. Kod xDSL-a, sa jedne strane je povezan sa telefonom, a sa druge je povezan sa modemom, ruterom, switchom (sve u jednom uređaju). Televizor ne može biti direktno povezan na ruter. Obično ispred televizora postoji stb (risiver), u njega ide utp kabl. Kod kablovske ima isto spilter, pri čemu kod kablovske jedna strana splitera vodi do stb-a (ulazi koaksijalni kabl, onda on ide na tv), a modem je utaknut u telefon. Pošto je televizor pasivan uređaj (on dobiva sliku od nekog uređaja i prikazuje je), ako hoćemo da naš tv može pristupiti mrežama (youtube i sl.), on posebno mora biti povezan sa ruterom (bežično ili kablom, ne preko stb-a). Za gledanje tv programa nam treba stb (preko njega dolazi slika).

#### Virtuelizacija

- Virtueleno je nešto što nije stvarno. Virtuelna realnosti - imam privid stvarnosti odnosno realnosti koja nije stvarna. Virtuelna memorija - programi imaju utisak da imaju pristup RAMu, a ustvari imaju pristup disku (laže im operativni sistem). Za onoga kome se stvara privid, on nema pojma da se radi o prividu.
- Virtuelni LANovi - ako imamo jedan switch, možemo reći 4 porta pripadaju jednom virtuelnom LANu (WLAN1), a druga 4 porta drugom LANu (LAN2), čime ćemo onemogućiti komunikaciju između ova dva LANa - od jednog fizičkog LANa pravimo 2 virtuelna LANa. Znači ako imamo jedan veliki switch, a trebamo da imamo 3-4 lokalne mreže, taj jedan fizički switch ćemo logički podijeliti na 4 različite mreže, i ponašat ćemo se kao da stvarno imamo 4 različite mreže (pri čemu se naravno treba povezati na ruter da bi ovo funkcionisalo)
- Virtuelne privatne mreže - mreže koje nisu privatne ali nam se čini da jesu. Za šta se ovaj pojam najčešće povezuje: npr. Ima firma koja se bavi razvojem softvera, ima svog uposlenika koji je otišao negdje preko svijeta i sjeo za PC, putem virtuelne privatne mreže on kad se zakači na svoju mrežu on ima utisak da sjedi u svojoj kancelariji (znači nalazi se u lokalnoj mreži svoje firme, ali se zapravo nalazi preko svijeta). Dakle mreža preko koje on komunicira je javna, ali podaci koje on šalje se šifriraju prilikom slanja i nikome na javnoj mreži nisu dostupni u čitljivom obliku.
- Virtuelne mašine
- Virtuelizacija mrežnih funkcija - ne samo da su sad računari virtuelni, sad su i ruteri virtuelni

### PREDAVANJE 3.2

#### Virtuelizacija nastavak

- Virtuelizacija nije došla sa virtuelnim mašinama nego mnogo ranije; Ideja je da nešto što je fizički predstavimo kao logički. Operativni sistem sa hardverom 'priča' preko drajvera (odnosno između je kontroler, OS priča sa kontrolerom, a kontroler sa hardverom tako da kontroler putem softvera može odgovoriti onom ko postavlja pitanje šta hoće -> on apstrahuje). Generalno, aplikaciju ne interesuje kako je npr. disk organizovan, bitno joj je koliko ima prostora i da može ispisati na disk (kao i RAM, za aplikaciju je nebitno da li je to virtuelni ili fizički RAM, bitno je da ga ima koliko je potrebno za tu aplikaciju).
- Šta bi bila virtuelna mašina? Mi smo OS-u dali hardver na kojem on može da se izvršava tj. stvorili privid da postoji računar koji je njemu na raspolaganju (sa memorijom, diskom i ostalim resursima). Ustvari, OS je dobio samo dio fizičkog hardvera na kom postoji. Virtuelizacija omogućava da u jednom trenutku na jednom računaru imamo pokrenutih više operativnih sistema. Bitno: Kada imamo OS na jednoj fizičkoj lokaciji, možemo ga premjestiti na drugi računar bez problema. Kako? Samo kompletnu virtuelnu mašinu prebacimo, ono što to omogućava je virtual machine monitor. Broj virtuelnih mašina na jednoj fizičkoj mašini zavisi od resursa koje ima taj fizički hardver i kakvi su zahtjevi virtuelnih mašina.
- Zašto su VM značajne? Kompletan cloud je podignut na VM tj. sve što se dešava na cloudu se dešava na VM. Većina servera danas su virtuelni, nisu fizički. Neke prednosti: troši se manje struje ako imamo jednu fizičku mašinu i više virtuelnih nego da imamo više fizičkih mašina; Problem hlađenja je veliki problem u data centrima - lakše je hladiti jednu mašinu s više virtuelnih na njoj nego više fizičkih mašina; Za povezivanje više fizičkih mašina potrebni kablovi, switchevi dok VM povezujemo virtuelno - nema kabliranja i zauzima se manje fizičkog prostora; Prebacivanje VM s jedne fizičke lokacije na drugu je kopiranje fajlova (virtuelna mašina je ustvari skup datoteka)
- Primjer: Odemo kod cloud providera i tražimo računar određenih performansi (cpu, RAM, disk). Tehničar neće složiti taj računar za nas, nego će u nekom softveru napraviti VM i

dodijeliti joj te resurse koje smo tražili i kad pristupimo tome imat ćemo privid da imamo te resurse

- U laboratorijama je na računarima instaliran Linux OS (Ubuntu), a unutar postojećeg OS instaliran Virtual Box, a unutar Virtual Box-a virtualne mašine tako da komunikacija VM s hardverom ide kroz VM monitor koji ide kroz host OS. Host OS naravno koristi jedan dio resursa tako da smo usporili procese na VM. Većina virtualizacija u data centrima se zasniva na nečemu što se zove hypervisor. To je softver, tj. virtual machine monitor, koji je instaliran direktno na hardver - možemo ga zamisliti kao mali operativni sistem, ni blizu složen kao OS koje mi koristimo koji su potpuno funkcionalni.
- Kontejneri: Jedna varijanta je, da bi osigurali željeno okruženje za aplikaciju, instaliramo kompletnu VM sa nekom aplikacijom; Kontejneri omogućavaju drugu stvar tj. sadržavaju aplikaciju i sve što je potrebno aplikaciji za njen rad; Prednost u odnosu na VM je što je to manje, nema kompletnu infrastrukturu kao OS nego samo ono što mu treba (slično kao apk kod android-a)

### PREDAVANJE 3.3

#### Instalacija operativnih sistema i softvera

- Prilikom instalacije OS-a mogu se instalirati komponente koje nam ne trebaju i koje će trošiti resurse, a neće se instalirati nešto što nam treba (tj. prilikom instalacije ne treba samo 'next next next' nego negdje treba stati i pročitati)
- Računari se generalno po namjeni dijele na serverske i klijentske. Serverski pružaju uslugu ne jednom, nego većem broju korisnika dok su klijentski namijenjeni samo za jednog. Na serverskim je obično instalirana jedna ili više usluga (web server, mail server i sl.) dok na klijentskim nema takvih usluga. Također, na serverske računare vrlo često nema potrebe da instaliramo grafičko okruženje jer njima nikad ne pristupamo direktno - jednom konfiguriramo i pustimo da radi u pozadini. Windows ima strogu podjelu na klijentske i serverske OS.
- Serveri obično ne dobijaju IP adresu od DHCP-a nego je mi podesimo ručno. Razlog za to je da serveri uvijek trebaju biti na poznatoj IP adresi, a ne da je jednom jedna, drugi put druga.
- Prilikom instalacije Linuxa pita nas za swap memoriju, to je virtualna memorija, pita za vremensku zonu. Potrebno je definisati privilegovanog korisnika, instaliraju se i određeni softveri, servisi (npr. DNS servis- aplikacija ne komunicira direktno sa DNSom nego se obrati servisu koji se obrati DNSu za pretvaranje domenskog imena u IP adresu). Svaki od tih servisa troši dio RAMa, možemo ugasiti one koji nam ne trebaju
- Kloniranje - u laboratorijama svi računari imaju instaliranu istu konfiguraciju. Ne koristi se virtualizacija na nivou hipervizora, nego imamo hostove. Instalacija više puta istog OS-a je dosadna i zamorna tako da postoje metode kloniranja koje omogućavaju lakšu instalaciju OS-a, a i softvera unutar OS-a. Ono što se danas radi je da imamo softver na dijeljenom računaru, tj. pristupamo njemu da bi koristili softver
- Licenciranje softvera - potrebno je definisati prava korištenja softvera. Microsoft softver dolazi u izvršnom obliku odnosno ne možemo ga mijenjati (nemamo njegov izvorni kod), ne možemo ga koristiti bez saglasnosti proizvođača. Postoje i druge liberalnije licence koje nam kažu da neki softver možemo koristiti, promijeniti, dati nekom drugom itd. Postoje i softveri koji su besplatni koje smijemo koristiti, a ne smijemo mijenjati. Postoji i licenca koja kaže da softver možemo koristiti, ali ako ga budemo mijenjali da napišemo koje su promjene urađene te da se open source dio ne smije koristiti u komercijalne svrhe
- Instalacija softvera: dobra praksa je instalirati softver na SSD (jer je brži), a podatke čuvati na HDD. Linux ima pored instalaciju izvornog koda, instalaciju ??? i instalaciju paketa. Svaka distribucija linuxa ima svoj format paketa. Paket je instalacioni softver koji ima u sebi definisano šta mu sve treba da bi radio i prilagođen je instalaciji na toj distribuciji. Preferirani način za instalaciju bilo čega je iz paketa. Instalacija izvorni kod- da bi instalirali nešto iz izvornog koda moramo imati kompajler jer od izvornog koda moramo napraviti izvršni.
- Kada upalimo računar: ako je računar uključen u struju osnovna napojna jedinica je pod napajanjem i ona daje napajanje matičnoj ploči. Šalje se signal matičnoj ploči da započne proces pokretanja računara. Na matičnoj ploči (ne na nekom od medija) se nalazi komad trajne memorije u koju je upisan softver BIOS (Basic Input Output System). Prva stvar koju uradi je provjeri šta ima od hardvera i da li je sve ispravno. Ako provjera hardvera prođe uspješno, nakon toga pokrene operativni sistem tako što prvo provjeri sve trajne medije sa kojih bi se OS mogao pokrenuti (hard disk, usb stick, cd, nekada floppy). U biosu je definisano kojim redoslijedom će se to provjeravati. Kada pronađe medij, onda on sa tog medija pokrene boot manager, a nakon toga OS. Kako da znamo kada je u pitanju problem, da li je hardverski ili softverski? Ako je hardverski bios će dati znak ili se OS



neće pokrenuti, ako je softverski OS bi se trebao pokrenuti

## **PREDAVANJE 4**

### PREDAVANJE 4.1

#### PREDAVANJE 3 - NASTAVAK

- Prilikom pokretanja os pokreće se dosta servisa
  - Procesi koji se izvršavaju po potrebi i pružaju različite vrste usluga drugim programima ili drugim računarima preko mreža
- Servisi su programi/usluge (unutar os) koji se pokreću s pokretanjem računara, ako ih ne trebamo a izvršavaju se, troše nam resurse
- U windows registrima upisano koji se servisi pokreću prilikom pokretanja
  - Procesi koji se izvršavaju u pozadini i pružaju usluge drugim procesima
  - Manuel, automatic
- U linux u runlevel editor, spisak servisa koji će se i neće pokrenuti
  - Imamo mogućnost da pregledamo taj spisak svih servisa
  - Offline files, matlab server etc
  - Preporuka proći skorz spisak servisa, odrediti šta nam treba
- Servisi odgovaraju procesu računara (objašnjenej task manager i mogućnosti koje imamo)
  - Netstat komanda u ubuntu i u windows koja pokazuje koje mrežne konekcije su otvorene s računara
  - Ako je računar bot, mi ne upravljamo njime, u mrežnoj konekciji se pojavljuje neka za koju nemamo pojma šta je
  - Možemo vidjeti koji programi se otvaraju pri startup
- Gašenje računara
  - Nemoj čupati iz struje jer diksovi se mogu oštetiti i os se može držati u kešu, čeka da se oslobodi memorija, ako nasilno ugasimo, može doći do oštećenja
- Nemoj čupati usb samo ako si pisao nešto po usb, da mu kažeš da si završio s njim
  - Ako si samo čitao s usb, može se bez safely remove

#### PREDAVANJE 4

- Kupimo novi računar, instaliramo os, on ima prostora, instaliramo softver
  - Entropija iz stabilnog stanja natrpa sve, Popravimo, konfiguriramo, povremeno ažuriramo sistem, Ponovo instaliramo os, u jednom trenutku cikne (ona slika sa trećeg slajda)
  - Potrebno je redovno održavati softver i os u konfigurisanom stanju
- Personalni i serverski hardver i njihove razlike
  - Kad se pokvarni personalni računar-pogađa jednog korisnika (obično ne tražimo čuda od njih, ne rade 24h etc)
  - Ako se hardver na serveru pokvari-pogođeni svi korisnicu na serveru (moraju biti pravljani da rade 24h, pouzdani)
  - Kad kuujemo računare(svejedno jel laptop ili pc) postoje 2 klase:

Kućni (imam disk takav i takav, ne znam ko je proizvođač,  
mogu ga personalizovati, održavati kako meni odgovara)

- Poslovni (skuplji, ali nisu puno bolji karakteristikama, kupuju se u više komada, znaju tačno koji disk kojih performansi pa su lakši za održavati jer svi istu stvar imaju)

- Serveri:

- Da bi dobili odgovarajuću pouzdanost servera, kućište je predviđeno da se može dodati više komponenti
- Serveri koriste drugačije procesore
- Ulazno izlazne komponente nisu miševi, ekrani etc , nego su su druge mreže? (ne čuje se fino koja je riječ)
- Server intenzivno komunicira s vanjskim sistemima
- Nadogradivost jer Ideja je da je životni vijek servera> životni vijek računara pa kad se nešto update u svijetu računara, da ne moramo kupovati novi nego nadograditi tu novu tehnologiju
- Stoje u komunikacionim ormarićima (kao u labu) jer mu možeš prići sa svih strana, standardizovane veličine etc
- Dodaci za visoku dostupnost (sve što se može pokvariti se može zamijeniti=
  - Puno diskova
  - Pravljen je tako da se sve može zamijeniti bez gašenja,
  - Rezervno napajanje
  - Sve što može biti duplo-jeste
  - Gledamo garanciju, dobar customer support
  - Često omogućava daljinsko upravljanje ( da se može upaliti i ugasi te užasenom serveru pristupiti preko mreže da se uradi npr dijagnostika)
- Serveri nekad pravljani kao veliki računar:jedan server-jedna funkcionalnost
- Pojavila se ideja više servera jefitinih i od njih napraviti konfiguraciju jednog servera
- Blade serveri:
  - Serveri inače danas liče na malo dužu spužvu (veličinom)
  - Nalazi se cpu, ram i nešto hardvera
  - Puno njih poredamo, na taj način dobijemo resurse
  - Kako ih objedinujemo? Da li na jednom serveru više virtuelnih mašina, da li ćemo sve te resurse udružiti u jedan logički?
  - Ono što se danas radi je kad imamo puno tih servera koji se objedine u jedan skup resursa na kojima se dignu viruelne mašine, pa sve usluge ustvari pružaju virtuelne mašine
  - Blade serveri su pravljani da ih bude više, napajanje i diskovi su odvojeni od servera
- Server sobe bi trebale biti
  - S ograničenim pristupom
  - Ako enstane struje da imamo napajanje backup
  - Treba imati kablova- sve će se pokvariti, treba imati dodatno
  - Statički podovi (protiv statičkog elektriciteta) treba biti moguće prići svakom serveru bez zapinjanje na kablove
  - Treba biti relativno očito šta je šta, ne samo administrator nego i ostali (ako se nešto desi administratoru), danas to ne znači puno jer se sad sve obavlja na VM
  - Treba biti zaštićeno od vode, od groma i ostalih nepogoda

#### PREDAVANJE 4.2

- Data centar
  - Većina internet događanja i mrežnog saobraćaja koji se dešava dešava se u data centru
  - Zamisli google, fb, koliki je njihov web server-nema tako velikog
  - Oni organizuju svoje servere kao skup računara koji su odvojene mašine koji sve skupa daju uslugu
  - Pošaljem zahtjev fb za nešto, on vjerovatno sa drugog servera šalje odgovor
  - Akamai firma koja pruža uslugu raspoređivanja sadržaja
  - Budući da treba pružiti uslugu ogromnom broju korisnika s jedne lokacije organizacija je:
    - To je s više različitih servera
    - Jedan frontend koji se bavi distribucijom servera tako da ne primjetimo da se to dešava, sve je seamless
  - Serveri uređaji su jako blizu međusobno, prave mali broj grešaka, postižu ogromne brzine

- Jedan data centar kako izgleda (slajd10)
  - Serveri nisu povezani kablovima, na zadnjem dijelu kućišta je mrežno ožičenje
  - Nazad imaju fiksno ožičenje koji svi idu u jedan switch koji ih povezuje međusobno
  - Svaki je povezan sa svakim pomoću neke strukture
  - Zahtjev naš s interneta ide prema nekom uređaju kojem odgovara zahtjev
    - Ako može 10 različitih servera opslužuje taj zahtjev –system bira takav da se stvara minimalno opterećenje
    - Ako mora jedan, nalazi se taj specifičan server i uzimaju se podaci sa njega
    - Kad server dobije zahtjev možda treba podatke s nekog drugog server da dopuni ono što ima, on to pokupi preko data centra s tih drugih servera
    - Pa onda se kombinuju podaci u odgovor i šalju se korisniku
  - Uvijek se pazi na redundantnost, uvijek sve u milion kopija
  - Data centri se grade u hladnim krajevima-ne dozvoljava pregrijavanje-hladi se, visoki krovovi iz istog razloga
  - Pravljeni da kamion može ući unutra i dostaviti opremu
  - Rijetko se dešavaju kvarovi, jednostavan način održavanja
  - Grijanje je ogroman problem, postoje čitavi sistemi za hlađenje
  - Veliki provajderi imaju nešto slično 'svom' internetu, njihove konekcije ne idu preko globalnog interneta nego preko tog drugog
  - Većina datacentara ima svoje hidroelektrane jer toliko struje troše
  - Ogromni prostori (12 fudbalskih igrališta)
  - Datacentar se sad kupuje in a box-sve je napravljeno, samo priključimo na šta nam treba
  - Primjer vrhunske tehnologije na jednom mjestu, tu rade tehničari na održavanju, ali inženjeri ih prave i dizajniraju
- Diskovi i particije
  - Sve što pišemo na disk pišemo na particije
  - 1tb disk dijelimo na 2 particije, time prenamjenjujemo taj disk
  - Nekad mi sami trebali praviti particije, sad većinom diskovi dolaze podijeljeni u particije koje mi možemo eventualno dodatno podijeliti
  - Računari dolatili u 2 particije-system i data particije, sad je 1 velika (zezancija da će doći na ispitu koliko particija ima naš računar)
  - Ideja je da svaka particija bude nezavisna: sistemska particija i particija s podacima. Ako se desi nešto jednoj, druga neće biti affected. Ssd sistemska particija-na njoj ima operativni system, a particija na mehaničkom hard disku koja radi drugi sistem
  - Puna particija sprečava da se radi u njoj
  - Veličina particije zavisi od potrebe
  - Pravimo particije ili ih kupimo
  - Ako pravimo
    - Koliko particija pravimo u disku (zavisi koliko bios podržava)- do 4 primarnih na windows i neke po potrebi.
    - 1.Korak : odredimo koliko želimo particija, te kažemo drajveru koliko želimo
    - 2.Korak : formatiramo tj omogućavamo da se podaci mogu pohranjivati na disk
    - Čemu služi formatiranje-za brisanje?-njegova je posljedica brisanje ali TO NIJE NJEGOVA NAMJENA. Formatiranje Omogućava da se podaci mogu pohranjivati u disk uspostavljanjem datotečnog sistema
    - Način kako su datoteke organizovane je datotečni sistem
  - Veza između particija i diskova: više particija nego diskova
  - ne vidi se na windows-particiju vidimo kao poseban disk
  - 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
  - Os treba biti na posebnoj particiji ili na posebnoj grani (ponovljeno puno puta!)
  - Korisničke datoteke obično odvojene (windows-users, linux : /home)
  - Sve stvari dobijaju podatke iz različitog izvora, mijenja se različitim tempom
  - Kod pravljenja backupa: možeš kopirati kompletan disk, najbitnije je praviti backup korisničkih datoteka-kopija samo onoga što ne možemo naći na drugom mjestu. Ako je dat sistem organizovani tako da ima folder koji se treba backup dnevno/sedmično etc to je idealno, jer onda samo njega backupamo
  - Datotečni sistemi su hijerarhijski organizovani
    - Odvajaju se korisnički direktoriji i proizvodi u razvoju(npr softver koji trenutno kreiramo)-najbitniji
    - Manje bitni tuđi softveri i lokalne baze podataka
  - Kako pristupiti dijeljenom folderu na drugom računaru?-preko mreže-potrebno je znati adresu računara (smb je čest protokol)- možemo dijeliti datoteke ili fajlove i možemo ih dijeliti na serveru ili na lokalnom računaru
  - Ako imamo dijeljeni folder potrebno nam je znati koji korisnici pristupaju njemu i kako da im se olakša pristup maksimalno (nećemo staviti folder namijenjen studentima u neki folder namijenjen profesorima)
- Računarska mreža
  - Zajednica svih korisnika koji imaju pristup toj mreži
  - Ako jedan računar zaražen, svi računari koji su povezani na tu mrežu su izloženi istoj opasnosti
  - To je zajednica prema kojoj se treba odnositi prema određenim politikama i pravilima ponašanja
- Mreža je zajednica-sve što radimo treba biti u skladu s pravilima ponašanja. Ako ne radi zajednica, ne radimo ni mi
- Morris worm
  - Worm/ crv je komad softvera koji se replcira na druge računare (pravi svoju kopiju)
  - Morris worm je kreiran u laboratoriji na jednom računaru, međutim procurio je s tog računara i toliko se razmnožavao da je skoro internet pao zbog njega
- Da bi se reguliralo ponašanje na računarima-
- Računari su obično dio neke lokalne mreže koji ima sistemsku politiku koja definiše kako se koristi mreža, šta su čije dogvoronosti, kako ko treba da se ponaša, šta se dešava u određenim situacijama, koji resursi se dijele u kojem obliku, kako se održava... Korisnici znaju na koje resurse mogu računati, a administrator zna i utiče na to koliko ko može trošiti (sve ovo određuje sistemska politika)
- Pitanje -da li je bolje imati sve iste računare ili raznolikost unutar neke organizacije?
  - Ako su svi isti- ako se pojavi greška na nekom os il uređaju vrlo je vjerovatno da svi imaju istu falinku, ali je lakše održavati, treba manje ljudi koji znaju radiit s jednim os/hardware/nebitno. Ako se desi problem factory- onda smo u problemu
  - Ako su svi različiti-na zavisi od problema sa factory issuse, međutim vrlo je teško održavati, treba puno ljudi da rade na tome
  - Većinom preovladava uniformnost sa određenom dozom raznolikosti da bi se izbjeglo gubitak svega ako to zakaže iz nekog razloga (tako da je to tačan odgovor)
- Umrežavanje
  - Prvi računari-ogromni-sva procesiranja sa mreže se vrši na centralnom računaru, korisnici imaju tasturu i monitor, koji omogućavaju slanje podataka prema mainframe, koji prikazuje sve primljene rezultate
  - Unix ideja distribuiranog modela-zašto bi jedan računar radio sve? jedan računar može i pružati i konzumirati usluge (podići servirs npr jedan računar web server, drugi je mail server i jedan drugom pomažu)
    - Računar može obavljati bilo koju ulogu
  - S windows mašinama imamo podjelu na serverske i 'korisničke' računare-mogu

- se koristiti jedan kao drugi ali nisu optimizovani za to
  - Macintosh ima slično unixu
  - Danas:
    - Peer to peer svaki pruža i peuzima-distribuirani model-ideja je imati zajednicu onih koji čine mrežu
    - Svaki čvor konzumira usluge drugih čvorova i pruža usluge tim čvorovima
    - Cloud computing-procesiranja na cloudu, naši računari su terminali-mainframe i cloud su nam ista stvar što se tiče naših računara
    - Centralno mjesto obrade podataka a korisnički računari služe za unos
    - Ima jako puno mrežnog saobraćaja koji se dijeli s ostatkom svijeta, i daleko je
    - Neko drugi održava, tako se razlikuje od lokalnih servisa. Od toga zavisi koliko energije trošimo na održavanje
- Mrežni servisi
  - Sve usluge koja mreža pruža su mrežni servisi
  - Web server, mail server-sve su mrežni servisi
  - Server je softver koji pruža uslugu davanjem podatka formatiranih u odgovor na zahtjev
  - Svaki server prihvata vrsu zahtjeva obradi ga i odgovori, to je proces koji se izvršava pozadinski i pruža uslugu
  - Na windowsu 'server' znači hardverski server
  - Kad imamo više korisnika na nekom računaru svaki želi sebi da prilagodi i imaju personalized korisničko iskustvo
    - Windows je bio sa jednim korisnikom, sada se korisnički podaci spašava u korisnike/users
    - Unix:sa '.' Su folderi- te datoteke opisuju sve parametre korisničkog okruženja
  - kod reinstalacije-gubimo korisnička podešavanja
  - Kod repair os- samo prepíše systemske datoteke sa ispravnim verzijama i ne dira korisnička podešavanja pa je bolja opcija od reinstalacije u većini slučajeva
- Moramo uzeti u obzir korisnike-oni generišu opterećenje sistema, oni su nepredvidljivi
  - Sistem treba biti projektovan tako da podnosi najveće opterećenje (međutim kad ima manje opterećenje to onda stoji neiskorišteno)
  - Zgodno je omogućiti da se omogući raspoređivanje korisničkog opterećenja u različitim vremenskim intervalima

#### PREDAVANJE 4.3

- Analiza mreže
  - Idealno je praviti novu mrežu
  - Češće moramo nadograditi postojeću mrežu moramo otkriti gdje vode kablovi
    - Način otkrivanja kako je mreža organizovana je poseban zadatak
    - Kako su čvorovi povezani, nije bitno kuda ide kabl ali gdje završavaju utičnice i kako 'oživjeti utičnice
- Kad mijenjamo računare, moramo znati šta mijenjamo
- Idealna situacija da postoji dokumentacija i da ima šema nacrtana s utičnicama, npr na fakultetu toga nema
- Snmp protokol koji treba da omogući da s jednog mjesta možemo poslati poruke raznim uređajima u mreži i dobiti odgovor
- Osnovni alati
- Računar ima (49 slajd) imena
  - Jedan od alata koji ćemo često koristiti alat za promjenu dns-a (nslookup komanda pomoću koje pravimo upit za domensko ime računara napiše nazivi adresu servera koju će on pitati )
  - Kako možemo znati koja je ip adresa webservera etf-a? nslookup\_ [www.etf.unsa.ba](http://www.etf.unsa.ba)
  - Alat za nalaženje servisa je to

- Kad šaljemo mail na etf.una.ba
  - Moramo znati koji se (9-11)
  - Mail servis radi tako da definišemo korisničko ime na domeni. Na dns zahtjev napišemo domensko ime
  - Možemo promijeniti mail server u svom domenu bez daikoga obavještavamo, postoji upit koji je mail service za enku domenu
- Posebni serveri su name serveri
  - Koji je server , tip upita ns i dobijamo onda name server ip adresu
  - Postoji mogućnost da kažemo nme serveru daj sve računare unutar domena
  - Ls list to previše info otkriva o nekom domenu
- Dig sličan ns lookup
  - -t a [www.etf.unsa.ba](http://www.etf.unsa.ba) vrati na toj ip adresi
- Alati kojima ispituje povezanost mreže
  - Ping služi za provjeru povezanosti nekog uređaja s drugim u nekoj mreži, također(windows 5 pingova i tjt , linux pinga do beskraja)
    - Možemo ip adresu ili domensko ime pingati
    - Vrijeme potrebno da ping ode to wwunsa ba i vrati se
    - Da li paketi mogu proći
    - Koliko vremena treba paketima (saznamo koliko je kašnjenje) 100ms trajanje je ok, 10ms skoro nemoguće
  - Tracing, tracert
    - Tracert(route) ona pokazuje da li imamo konekciju nego pokazuje domenska imena svih rutera kroz koja je paket prošao 20-tak se obično prođe rutera
    - Vidimo gdje se javilo kašnjenje,Vidimo gdje se nalazi problem između nas i greške konekcije
    - Kojim putem paketi idu
    - Radi windowsu i na linuxu
  - Odzivi zavise od servera, ako smo mi na telemach a kolega na bhtelecom, vrlo dugo traje iako smo fizički blizu
  - Unix ima grafičko okruženje za ping i za traceroute etc
- Šta od mrežnih servisa u našoj mreži postoji
  - Nmap alat može od skupa adresa uraditi pretragu mrežnih servisa
    - Kako pristupam web serveru, moj računar pošalje na port
    - Softver za izabrani skup ip adresa pošalje paket na sve portove. Ako an račuanru ima web server koji sluša na tom portu to ćemo saznati preko njega
    - Skenira mreža, ako se neki port javi, tu ima servis, na osnovu odziva saznajemo koji je to servis
    - Nemoj na mrežu koja nije naša (to je kao da uđeš u zgradu i ulaziš u svaki stan)
- Usposobljavanje mrežnih servisa
  - Moramo odgovoriti na sva pitnaja 65 slajd
  - Bitna nam je svrha
  - Koje usluge, kakvo je trenutno stanje, šta mogu sad
  - Nekad je bila ram ,
  - Jedan računar sa svim servisima ili po jedan servis na puno računara
  - Prilagodimo se onome što imamo, i učinimo ga takvim da s emože prilagoditi
  - Ako nudimo korisnicima moramo imenovati servis (url http(protokol)://www.etf.una.ba(domensko ime)/host(putanja))
  - Ako se radi s datotekama onda hijerarhijski se unosi
  - Alias alternativna imena (mail servis se zove webmail, zimbra, a njegovo originalno je ime igman)
  - Web server treba da stoji u vanjskoj mreži, trebamo mi pristupiti njoj
  - Servis koji pruža unutrašnjim korisnicima treba stajati unutar naše mreže, servis koji pruža uslgu vanjskim korisnicima treba stajati u vanjskoj mreži
  - Međuzavisnost-ako pružam uslugu a dns ne radi, nema poente nikakve
  - Nemoj se igrati na računarima koji imaju važne stvari ili vrše važnu ulogu, ima starijih, 'nevažnih', izoliranih iz mreže za igranje i testiranje

## PREDAVANJE 5

- DNS- pretvaranje domenskih imena u IP adrese
- Otkucam u web pregledniku npr. [www.google.com](http://www.google.com). Da biH dobili ip adresu: prvo se browser obrati dns resolveru (servis na operativnom sistemu) koji pročitao ip adresu dns-a, onda se on obrati name serveru čija je ip adresa definisana u postavkama računara. Onda se šalje name serveru domensko ime i on vraća IP adresu.
- Kad se šalje mail, on se šalje na ip adresu mail servera, npr. gmail servera i u ovom slučaju se isto koristi resolver.
- Koliko ima domenskih imena na svijetu ? Red veličine stotine miliona.
- Prvi name server kojem se resolver obraća se zove lokalni name server.
- Lokalni dns ne zna sve adrese nego se obraća root dns serveru.
- Formalno na kraju svakog domenskog imena treba da stoji tačka.
- Osnovnih domena (TLD-top level domena): ima oko 1000
- GTLD (generic top level domains)- napravljene kad je tek napravljen internet
- Ko upravlja domenskim imenima, ip adresama i protokolima? ICANN-Internet Corporation for Assigned Names and Numbers, imali ugovor sa američkim ministarstvom odbrane i postala je nezavisna organizacija
- To se radi kroz regionalne registre. Nama zanimljiv je regionalni registar za evropu (RIPE, od francuskog skracenica).
- ICANN podijeli sve na 5 svjetskih regiona, a onda se RIPE-u obrati ISP i kaže daj mi skup adresa.
- Slično je za domenska imena, pri čemu su domenska imena podijeljena na dvije grupe: GTLD (com, edu, gov, mil, net, org, arpa) i ccTLD-country code top level domain- ima iso standard koji pravi dvoslovne skraćenice za države
- Može se kupiti domena ali nikad se ne može biti vlasnik domene nego korisnik domene.
- Npr. hoću da kupim domenu .saša. Odem kod ICANN-a i platim za tu domenu, od koje treba da napravim biznis ili ne. Ja onda to mogu da izdajem ili šta god radim sa tim.
- U svakoj državi ima organizacija koja je sklopila ugovor sa ICANN-om kojim je ona ovlaštena da upravlja domenskim imenima ispod tog imena. Kod nas je to UTIC, on izdaje domene unutar .ba.
- Ko je vlasnik .ba domena? Je li ICANN ili država Bosna, s obzirom da ICANN sutra može obrisati čitav domen?
- Kompletan skup domenskih imena izgleda kao stablo. Imamo root, pa iz njega .edu, .com, .ba.....
- Kako izgleda hijerarhija ispod je u nadležnosti organizacije koja ga vodi, npr. UTIC. Ima samo ograničenje na ukupnu dužinu i ograničenje na broj nivoa poddomena. Bilo je ograničenje da se koriste samo simboli iz ASCII tabele, ali je to danas ukinuto. Mogu biti arapska slova, ćirilica, itd. Sad npr. Srbija pored domene .rs ima domenu .srb(ćirilicom), kinezi imaju neke svoje domene, itd.
- Hijerarhija je jako bitna.
- Vratimo se na pitanje od kojeg smo krenuli (tražimo [www.etf.unsa.ba](http://www.etf.unsa.ba)). Naš name server je dobio pitanje za neki server koji se nalazi duboko u hijerarhiji. Pitamo prvo root name server. Odkud njemu adresa root name servera? Ide kroz hijerarhiju gore da dođe do adrese, jedna varijanta koja je donekle tačna. Formalno u name serveru, koji je ustvari komad softvera, u konfiguraciji ima adresa root dns servera. Druga varijanta koja može biti je da on pored ovoga ima upisanu drugu adresu, u slučaju da lokalni dns ne zna tu adresu, nemoj odmah bihuzurit root nego pitaj prvo npr. UTICov server, bhtelecom i slično. Adrese ovih servera se moraju upisati jer on nema načina da ih sazna.
- Ima 13 root dns servera. Šta zna root server? One ne zna sve, nego samo jedan sljedeći korak. Dat će ip adresu name servera top level domene. Npr. pitam ga [www.etf.unsa.ba](http://www.etf.unsa.ba), on kaže ja ne znam šta je ovo sve ali znam ko će ti pomoći i proslijedi adresu .ba servera.
- Dalje pitam .ba server, koji me dalje proslijedi na sljedeći name server u hijerarhiji, [uns.ba](http://uns.ba). Pa opet njemu šaljem upit, on mi vrati ip adresu name servera [etf.unsa.ba](http://etf.unsa.ba). on će mi onda vratiti ip adresu [www.etf.unsa.ba](http://www.etf.unsa.ba).
- Preslikavanje domenskih imena u ip adrese se zapisuje samo na jednom mjestu, unutar name servera i on se naziva autoritativni server.
- 13 root servera su svi potpuno identični. Imaju potpuno iste podatke i rijetko se to mijenja. Da se sruši ovih 13 servera, stao bi internet.
- Dakle da sumiramo:
  1. Kucam u browser [www.etf.unsa.ba](http://www.etf.unsa.ba)
  2. On se obraća operativnom sistemu tj. Resolveru
  3. Resolver je pročitao ip konfiguraciju i dobio ime dns servera
  4. Poslao je dns upit
  5. Dns se obratio root name serveru



6. Root je vratio ip adresu za .ba domen
  7. On je vratio ip adresu za .unsa.ba
  8. I on je vratio .etf.unsa.ba
  9. Name server je to vratio resolveru
  10. Resolver vratio browseru
- Jel uvijek mora preko roota da se vraća? Postoje dvije vrste upita: iterativni i rekurzivni
  - Rekurzivni: Iz ugla resolvera, on je pitao name server i on mu je dao odgovor, nije ga šaltao haj tamo haj vamo. Name server uvijek daje klijentu odgovor (mijenja se ko je klijent ko je server, npr. klijent je dns za .ba a klijent je dns za .unsa??? ovo sam ja skontala valjda je dobro). Na kraju se resolveru vrati kompletan odgovor.
  - Iterativni: Server neće odgovoriti sa konačnom ip adresom nego će mu poslati ip adresu sljedećeg servera koji mu treba, u formatu upita (zahtjeva u paketu koji označava dns upit). Onaj koji šalje upit naznači hoće li iterativni ili rekurzivni upit.
  - Iz ugla roota: zamislimo da kompletan internet pita onih 13 root servera da im resolva. Vjerovatno bi internet stao. Iterativni način je bolji jer se rastereti root.
  - Kesiranje: Web preglednik ima svoj dns cache. Ako jednom tražim neku stranicu, on će jedno kraće vrijeme držati to preslikavanje. Resolver ima svoj cache. Name server ima cache, on može imati ip adresu za unsa.ba pa on nema potrebe da pita root. Dobar dio upita se odgovara iz keša. Kad dobijemo odgovor iz cache, pomocu nslookup, na njemu piše: ne autoritativan odgovor znači nije došao do alternativnog servera. Odgovor nije došao direktno sa etf.unsa.ba gdje je on originalno zapisan nego već zna da ima to negdje.
  - Dinamički dns: ako sam ja korisnik, svaki put kad se zakačim dhcp mi dinamički dodijeli ip adresu a ja hoću nešto da hostiram na svom računaru. obratim se dinamičkom dns-u koji mi da neko domensko ime. To domensko ime se veže za ip adresu koju trenutno imam. Znači ne treba da se kešira ovo preslikavanje.
  - Vrijeme keširanja je definisano kad pitam nekoga. Kad je autoritativni name server odgovorio serveru koji ga je pitao, on je u odgovoru poslato ttl (time to live). To je vrijeme za koje on očekuje da se ništa neće promijeniti. Uobičajeno vrijeme je 1 dan.
  - Npr., kad je etf prelazio sa bhtelecoma na logosoft, promijenili su javne ip adrese. Trebalo je 48 sati da se svi keševi promijene.
  - Kad se nesto rjeđe mijenja stavi se duži ttl. Ovime se smanjuje opterećenje na infrastrukturu.
  - Do sad smo imali na svakom nivou hijerarhije jedan name server.
  - Hoćemo sada da otvorimo stranicu www.ri.etf.unsa.ba. Da li ri mora imati svoj name server? Ne mora. Postoje zone koje mogu uključivati jedan hijerarhijski nivo a može i više.
  - Mogu biti dvije varijante. Prva da imam unesena preslikavanja za ri.etf.unsa.ba i druga da je unesen ip nekog dns severa koji je zadužen za ri pa preusmjerim njemu.
  - Kako izgleda baza dns podataka? Kad autoritativni name server odgovori, postoje tačno formati kako izgleda odgovor. Ono što postoji u bazi su zapisi o resursima (resource records-rr). u njemu piše domensko ime, ttl, klasa, tip zapisa i vrijednost. Najvažniji su zapisi tipa A, pretvara ip adresu u ime. Ova baza je bukvalno tekstualna datoteka gdje pišu ovi zapisi.
  - ns se uvijek dodaje na ime domene.
  - Postoji datoteka gdje se vodi evidencija onih 13 root servera.
  - Otvoriti mapu servera u predavanju. Svi F name serveri imaju istu ip adresu. Rutiranje je podešeno tako da će te ruter proslijediti do najbližeg.
  - BIND softver se koristi na 10 od onih 13 servera. On podržava da imamo više servera: master i slave. Slave preuzima ako master otkáže, ili se može rasporediti opterećenje. On ima jedan direktorij - named ili dns. Unutar njega nalazi se direktorij master i direktorij slave. Unutar mastera se nalazi datoteka koja se zove kao zona odnosno domena.
  - Postoje reverzni dns upiti gdje pretvaramo ip adresu i domensko ime.
  - named.cache - datoteka gdje piše gdje se nalaze root serveri
  - named.conf - kako se zove zona i gdje se nalaze podaci o root name serverima, localhostu i reverzibilno mapiranje
  - Kad se name server upali, tj. BIND softver se pokrene on pročita named.conf fajl i vidi za koju zonu je zadužen i sve ostale podatke.

## **PREDAVANJE 6**

- Active directory - neka vrsta baze podataka u kojoj se vodi evidencija o korisnicima domena, računarima, grupama, itd. U suštini neki zapis o svemu što se nalazi na domeni.
- Active directory je Microsoftova izvedba LDAP-a (Lightweight Directory Access Protocol).

- Yahoo je jedan način pretraživanja. Način kako su oni organizovali pretraživanje je bio directory (imenik).
- Kako Google pretražuje? Kako zna da je neka stranica relevantnija od druge?
- On ima ogramnu bazu rijeci, recenica, fraza i slično. Kad ja nešto ukucam on ode u tu bazu i dobije redoslijed stranica za tu riječ. Taj redoslijed se mijenja otprilike jednom mjesečno, mijenja se algoritam koji to optimizira. Npr. ja mogu imati stranicu o igricama ali da se puno spominje dijabetes i da onda kad neko pretražuje dijabetes izađe moja stranica, iako nema veze sa tim. Nikad covjek iz gugla nije otvorio stranicu i rekao aha to je stranica o tome jer se sve dešava kroz algoritam.
- Imenici su zasnovani na tome da ljudi posjete stranicu i svrstaju u neku kategoriju. Ako u gogl ukucam "rezultati", dobiću puno nepovezanih stvari. Ono što su imenici omogućavali je da izaberem kategoriju, npr. sport i onda tražim rezultate. Time se povećala relevantnost rezultata koje dobijem.
- Kad nešto promijenim na stranici, potrebno je neko vrijeme da se to promijeni u svim keševima.
- Koja je razlika između imenika i baze podataka? Baza podataka je sve u što se stavljaju podaci, znači i datoteka je neka vrsta baze. Mi govorimo o relacionim bazama podataka, imenik možemo upisati u relacionu bazu.
- Jedna varijanta da napravimo imenik bi bila da napravimo relacionu bazu, ali mi onda treba softver koji će pojesti pola RAMa i pola hard diska. Jednostavnija varijanta je softver koji povezuje imenik. Active directory je zapravo imenik svih sadržaja koji se nalaze u imeniku, što uključuje korisnike, podatke o korisnicima koji uključuju korisnička imena, lozinke, itd. Postoji posebna klasa softvera koji su namijenjeni za pretraživanje i rijetko su ažurirani.
- SSO-single sign-on znači da se prijavim na jednom mjestu i onda za sve servise kojima mogu pristupiti sa tom prijavom se ne moram odjaviti (npr. kad se prijavim na etf mail, djelimično). Na zimbri se nalazi LDAP, imenik u kom su pohranjeni svi podaci. Ako hoću da promijenim šifru, to radim na zimbri i to se automatski promijeni za zamger, c2, jer su svi vezani za isti imenik.
- Dakle, imenik je kao neki repozitorij podataka koji koriste različite aplikacije, najčešće je to jedna tabela.
- DAP-directory access protocol - objasnio kako aplikacije mogu tražiti od imenika podatke, kako on vraća podatke
- Active directory je ustvari imenik kojem se pristupa putem LDAP-a.
- DIT (directory information tree) - stable kore predstavlja sve unose u imeniku
- DN (distinguished name) je jedan unos u imenik, zapravo adresa unosa u imenik koje treba da je jedinstvena.
- DNS nije pravi LDAP ali je fin primjer. Pogledaj u prezentaciji sliku hijerarhije. BA je jedan distinguished name i bitno je da na tom nivou u hijerarhiji nema isto ime. Na drugim nivoima može biti. Postoje komponente koje određuju hijerarhiju (organization (unsa), organizational unit (etf), country (ba), itd.)
- LDAP ne propisuje imena komponenti, koliko ih ima i slično, (možemo svoje izmisliti) nego samo daje uobičajnu konstrukciju.
- Unutar jednog DN imamo attribute i njihove vrijednosti. (npr. unosim Sašu a njegovi atributi su broj telefona, email, titula, itd.). Možemo imati razne objekte sa različitim atributima.
- Šema u ovom slučaju definiše sve vrste objekata koji se pohranjuju (studenti, profesori, prostorije, i slično). Rezervisanje prostorija se vrši preko LDAP-a. Kad unosim novu prostoju, specificirano je tačno koje podatke moram unijeti.
- Dakle, šema definiše klase objekata. Objekt je unos u imenik. Prilikom definisanja objekata definišemo koji su njegovi atributi. Jedan atribut može imati više vrijednosti. Za atribut se treba definisati njegov naziv, opis, koji je tip, koje su mu dozvoljene vrijednosti, maksimalna dužina i obično ima jedinstveni identifikator. Tabela uobičajenih atributa ima u prezentaciji.
- Način razmjene podataka je LDAP data interchange format. Prva linija definiše distinguished name i ispod njega nabrojimo sve attribute. Tako možemo uraditi efikasan eksport dijela stabla.
- LDAP možemo koristiti za evidenciju, pretragu, promjenu unosa, čuvanje lozinki.
- Kerberos je algoritam koji se koristi za potvrđivanje identiteta, provjerava da li je korisnik ovlašten da pristupa podacima.
- Zahtjev za povezivanje sa bazom - bind request
- Prije ovog zahtjeva LDAP može tražiti potvrđivanje identiteta ali i ne mora ako je pristup otvoren.
- Parametri za search zahtjev: base - treba da specificiramo odakle u hijerarhiji se kreće pretraga (npr., trazi unutar .unsa).

- Scope - govori dokle da ide (do kraja, samo na ovom nivou, samo na pocetku, i slicno).
- Veličina - ako ima 100 zapisa koliko će se od njih vratiti.
- Vrijeme - koliko dugo može trajati upit
- Atributi - ako npr. tražim samo mail studenta, ostali atributi mi ne trebaju, znači vrati mi samo mail
- Attrsonly - hoće li vraćati samo naziv atributa ili i vrijednost
- Search\_filter - npr. hoću da pretražujem sve studente koji se zovu Saša, omogućuje da se rezultat suzi
- Rezultat pretrage je zapis koji sadrži jedan distinguished name i svi atributi, plus još kod koji govori da li je operacija uspješna ili nije.
- Kad neki klijent hoće da pristupi: 1. otvara konekciju sa serverom, 2. potvrđuje identitet, 3. uradi operaciju koju je htio, 4. zatvara konekciju
- 1. Otvaranje konekcije: ldap\_bind funkcija koja ima povratne vrijednosti: ldap\_success i ldap\_errno
- Kad nešto pretražujemo ide funkcija ldap\_search\_s, zadnji parametar je &mesg, odnosno pokazivač na rezultate. Radi tako da samo ide redom i čim pročita jedan zapis, vrati ga kao rezultat pa ide dalje.
- Active directory je dobar jer se može napraviti neka administrativna politika koja se odnosi na sve što je u domeni. On isto prati kad se korisnik prijavio, da li ima privilegije. U principu on služi da se pohrani sve što ima u domeni.
- Napravljen je da se može prilagoditi i malim i velikim firmama, da se olakša administracija jer se svim podacima pristupa sa istog mjesta.
- Active directory nudi APIje koji omogućuju da iz drugog programa pristupamo podacima.
- Koristi DNS za imenovanje i lociranje servisa, znači ako ne radi DNS, ne radi ni active directory.
- Kerberos - algoritam/protokol za potvrđivanje identiteta. Jedna varijanta prijavljivanja na neki servis je da ukucam lozinku i računar provjeri tu loziku. Drugi način je ako imamo veliki broj servera. Kako onda da se prijavim? Da li svaki od tih servera treba da ima podatke? Ako promjenim loziku na jednom serveru kako onda na ostalim da promjenim? Kerberos omogućava da se ja prijavljujem na neki servis i kada on provjeri lozinku, da mi token. Na osnovu tokena ja mogu otići na drugi server i reći da mi je to dao ovaj prvi server. To omogućava siguran način prelaska na druge servise. Izda mi se neka potvrda, neki digitalni niz bita, koji ja pošaljem svim drugim servisima koje hoću da koristim.
- Osnovna jedinica active directorija je domen. Unutar domena imamo organizacione jedinice koje mogu biti hijerarhijski organizovane kao i domeni. Unutar toga imamo razne stvari, ljudi, računari, grupe i drugih resursa.
- Domeni mogu biti organizovani u stabla ili u šumu.
- U active directory se inicijalno upisuju korisnici, grupe korisnika i računari. Kasnije je dodano bilo šta, uključujući distribucione liste i sistemske politike (group policy object). Group policy object je način da kažete npr. nijedan korisnik na ovom domenu ne može pokrenuti neku aplikaciju, svi korisnici na domenu moraju imati takvu lozinku. Dakle, definiše sva pravila ponašanja na sistemu.
- Active directory omogućava da u njega pohranite bilo šta. Dok god druga aplikacija zna pročitati taj fajl to je uredu.
- Kompletan struktura active direktorija je unesena u šemu. Ona govori od čega se sastoji active direktorij. Šema se koristi za sve domene koje čine jednu organizaciju. Sistemske politike definišemo na nivou jednog domena.
- Active directory se fizički nalazi na kontrolerima domena. Da bi imali domen moramo imati računar koji se zove kontroler domena na kojem se nalazi active directory.
- Unutar domena se mogu praviti organizacione jedinice (Ako je domen etf onda organizacione jedinice mogu biti smjerovi, ili npr spratovi). Sistemska politika se primjenjuje na svijetu a može se definisati i posebno za neke.
- Domen se može organizovati kao stablo, pri čemu mora postojati glavni domen i onda poddomeni, slično kao dns.
- Suma nastaje kad se dva domena stave u root, znači dva na nivou hijerarhije. Oni predstavljaju dva potpuno različita prostora. Bitno je da sve što se definiše u jednom domenu je automatski prepoznato u drugom domenu. Sve je stvar organizacije, npr. svaki fakultet na univerzitetu bude jedan domen a mi svi skupa činimo forest.
- Bitna stvar kod domena je da fizička i logička struktura ne moraju biti identične. Fizička govori gdje se nešto nalazi. Npr. bhtelecom ima više svojih lokacija pri čemu se jedan domen može nalaziti na više lokacija u Tuzli u Sarajevu itd. Znači ne moraju biti fizički jedna pored druge, samo su povezane brzom konekcijom. Omogućava se razdvajanje logičke i fizičke strukture a opet da se sačuva informacija u fizičkoj strukturi.

- Nekad je bio jedan primary domain-controller a ostali sekundarni. Danas ima više ravnopravnih domen kontrolera. Jako je korisno imati više domen kontrolera jer ako izgubimo podatke sa njih, izgubili smo sve. Ako nemamo podignut dns, ništa neće raditi.
- Practical način za upravljanje korisnicima je preko grupa. Npr profesor hoće da podijeli materijale za ARM. Može ići jedno po jedno i dodavati privilegije svakome od nas. Drugi način je da napravi grupu i onda lako mijenja privilegije.
- Microsoft je napravio varijantu da se može prijaviti na svoj računar (što mi svi radimo) ili da se prijaviš na domen. Prijava na domen podrazumijeva da ako npr imam 4 servera neka i ja se prijavim na active directory odnosno domain controller, tada postajem član domenske grupe. Da bih mogla imati privilegije na lokalnom računaru, pravi se kopija globalne domenske grupe u lokalnu grupu jer računar uvijek provjerava lokalne grupe.

## PREDAVANJE 7

- Koliko ima websajtova na svijetu-skuplja se internet- br aktivnih sajtova se ne smanjuje ali broj host sajtova se smanjuje
- Web serveri-najpopularniji je apache-na globalnom internetu
- Microsoft se pojavio kao ozbiljan takmičar-firme
- Engine x- jako brz(bitno na opterećenim sajtovima)-profesionalne
- Google koristi svoje servere 2% interneta
- Koliko server radi nepotrebnih stvari
- Optimiziran web server za lokacije koje imaju veliki promet

Može biti jedan veliki server ako instaliramo web server

- Cloud web server
- Da li nam treba server koji se nalazi kod nas? Da li su naši korisnici lokalni?-sve to je potrebno definisati
- Stvar pouzdanosti, lokacije, napajanja etc -zato je server etfa u njemačkoj
- Da li je standardan pristup, ko ima pravo pristupa, koji softver ? Također potrebno odgovoriti prije pravljenja
- Apache kao skoro najpopularniji web server provajder
  - Open source znači da je on besplatan, te da možemo iskoristiti taj kod da nešto možemo naučiti, legalno ga je preuzimati
  - Možemo skinuti na zarne načine: uz izvršne datoteke, uz pakete ili izvorni kod i postaviti ga (instalirati)
  - Konfiguracija ima verzije i za linux i windows
  - Možemo ranije verzije skinuti po potrebi
  - Najbolje ako linux preko distribucije (sudo apt-get install apache)
  - Za windows najbolje instalacija izvršnih datoteka
  - Podešavanja
    - /etc/services datoteka u kojoj se nalaze definisana preslikavanja između usluge i porta na kojem se pruža
    - Httpd.conf(ili apache2.conf )- konfiguraciona datoteka koja može biti podijeljena na više različitih datoteka
    - Sve te konf datoteke u linux jako dobro opisane-jednostavne za čitati
  - Konfiguracije koje možemo da mijenjamo
    - Server route: gdje se datoteke nalaze, možemo odabrati šta želimo
      - Najbolje je napisati tu mail na koji se može javiti ako se desi neki problem
      - Da bi neki servis rezervisao port koji je manji od 1024 na linuxoidnim sistemima (ako hoće da osluškuje na port 80) web server se obraća operativnom sistemu da mu rezerviše port
    - Apache User
      - Da bi radio na serveru mora biti privilegovani korisnik
      - Ako se nešto desi sa web serverom neko provali ili nešto slično, u tom slučaju taj softver ima privilegije root te može praviti problem
      - Ideja je unutar apachea da se pokrene kao root na početku (jer mora). Nakon pristupa portu predaj svoje root privilegije i koristi se drugim korisnikom koji nije root (kao neinteraktivni korisnik npr)
      - Neinteraktivni korisnik-ne postoji lozinka, ne mogu se prijaviti kao taj korisnik, većina servisa u linux se samo kao

neinteraktivni korisnik

■ Keep Alive

- Web je prije bio puno više samo pisano. print pošalje stranicu, dobije sadržaj i konekcija se zatvori, to je često bio samo tekst pa je brzo išlo
- Kad su web stranice počele koristiti puno slika i sl. koji su na href, treba se zahtjev za sve te slike. Ako se konekcija prekine poslije svake slike, neefikasno je, u verziji http1.1 dodane perzistentne konekcije koje rade:
  - Klijen pita servera, server odgovori, server pretpostavlja da će biti još pitanja.
  - Keep alive je opcija koju biramo da kažemo da nećemo odmah prekidat konekciju nego čekati x zahtjeva
  - Možemo i postaviti vrijeme nakon kog server skonta da neće dobiti novi zahtjev, prekida konekciju i prekida server

■ Prava prisupta

- Za sve direktorije prava su restriktivna
- Niko nema pravo pristupa nijednom folderu unutar servera, zabranjen pristup svim stranicama za svaki slučaj

■ Virtual host (ovo je vrlo čudno objašnjeno na predavanju, puno pisanja po tabli, meni nije jasno pa su takve i bilješke, ovaj dio bih učila s nekog drugog materijala)

- Služi za (objašnjenj na tabli, ne kontam kako se desilo kako se šalje zahtjev)
- Ako ja želim da s web aplikacije poslužujem 2 sajta odjednom, to mogu uraditi ip adresa, port, kad doviye get vrati doc u get zhtjevu
- Virtualni host omogućava više stranica odjednom, kad dođe get zahtjev(trazi web podatke), webs server zaključi da
- Za svaki host koji se poslužuje definisana je putanja
- Kad dobiješ zahtjev z awww.arm.ba/folder1
- Ako nemamo virtuelnih hostova, apache treba d azna koji sajt poslužuje
- Ako više virtuelnih hostova
- Kako zna na koju web lokaciju zna da pošalje zahtjev ako piše web adresa i port?
- Koji objekat i sa koje od lokacija-to se nalazi na aplikativnom sloju
- Na istoj adresi na istom portu može biti više lokacija ali moramo definisati šta je root

○ TREBALO BI SE OVO NAUČITI ODNEKLE DRUGO JER JE PUNO PISANJA PO TABLI, NIJE BAŠ JASNO (15-25)

○ Pravila pristupa se uglavnom ne mijenjaju, dozvoliti svima pristup, ili neki dio stranice se može restrict

○ Microsoft ne nudi ovo-korisničke web stranice ~ -omogućava da korisnici web servera imaju svoje web stranice

- Definiše se komandom userdir i poziva sa weba npr\_ [www.etf.unsa.ba/~saša/](http://www.etf.unsa.ba/~saša/)

○ Uloga web servera da poslužuje objekte (objekat=datoteka)

- Web preglednik treba u get zahtjevu navesti koji objekat želi (putanja i naziv objekta) ali kad ww.nekisajt.ba (ako se ne navede objekat koji se traži)-ona tražimo da se otvori indeks -ako nije definisan objekat posluži index.html(ili neki sličan )

○ ScriptAlias-prije je većina interaktivnosti se pomoću skripti radila, sad nije toliko popularno

- U usr/local/apache2/cgi-bin/ folderu
- Da se skripte ne bi morale nalaziti na istom mjestu gdje ostale datoteke
- Odvajali smo skripte jer su se često koristile za druge stvari

○ Mime.types

- Kako web browser otvori sve tipove dokumenata (pdf,epub, docx...)?Zovne word/epub manager/bla bla da otvori dokument. Kako zna

šta treba zvati?

- Mime.types to omogućava
- Get zahtjev navodi objekat : 'daj mi objekat film.mp4'
  - Web preglednik nema pojma šta je to
  - Web server u svom fajl sistemu nađe tu datoteku (on ne tumači ništa, samo čita i vraća nazad u nekoliko paketa tom web pregledniku),
  - Da bi web preglednik znao protumačiti taj niz bajta, postoji content-type header gdje server u sklopu odgovora kaže koji tip je ta datoteka
  - Web server onda ima taj nastavak i zna šta da radi s tim
  - U konfig servera ima mimetypes datoteka, server pogleda mp4 u mimetypes i sve što piše o toj ekstenziji prepíše u taj header
  - Možemo definisati neku vrstu datoteke tako da on radi
- IIS (Internet information services)
  - Sve što smo morali za apache moramo i za ovaj potaviti.
  - Windows sve kroz grafičko oružanje-ako slučajno nema, onda može config
  - Moramo definisati default-odakle se kreće
  - Možemo wait, možemo drugu web lokaciju, sve kao i na apache
  - Sve je na slikama na predavaanju
  - U default document podešavamo šta ako ne pošaljemo path nego smao [www.nešto.ba](http://www.nešto.ba)
  - Web server se pokreće kao servis
  - Koristi mime types također, možemo dodavati svoje ekstenzije