INFORMACIJSKA TEHNOLOGIJA

Namen, vloga in pomen informacijske tehnologije

Informacijsko tehnologijo **uporabljamo** v:

- **bančništvu**: SWIFT – dogovor o označevanju bank, IBAN – mednarodna bančna številka računa, bankomati, spletna banka in spletno poslovanje;
- **telefoniji**: optični kabli, brezžična telefonija – LTE, 5G itd.;
- **trgovini**: črtni zapis EAN, spletne trgovine;
- **zdravstvu**: informacijski sistem s podatki o pacientih, računalniška tomografija (rentgensko slikanje), ekspertni sistemi in AI pri diagnosticiranju;
- **pisarniškem poslovanju**: urejanje besedil, finančno poslovanje itd.;
- **industriji**: računalniško vodena proizvodnja, robotizacija, CNC stroji itd.;
- **založništvu in tisku**: digitalni tisk in elektronski mediji.

Ravni uporabe IT so naslednje:

1. **računalniška pismenost**: znanje, ki omogoča samostojno, učinkovito in uspešno uporabo računalnika;
2. **razbremenitev pri delu**: delo z IT nam ne predstavlja napora, uporabljamo jo smiselno;
3. **izvajanje novih aktivnosti**: pri pisanju besedil uporabimo preverjanje črkovanja, preoblikujemo in urejamo slike, analiziramo podatke;
4. **intenzivna uporaba znanja**: ekspertni sistemi, inteligentna analiza podatkov, pomoč pri odločanju itd.

Alan Turing je teoretično zasnovo modernega računalnika prvič opisal l. 1936 – Turingov stroj. **Elektromehanični računalniki** so temeljili na elektromehanskih stikalih (relejih), bili so težki, veliki, zaradi mehanskih delov pa so bile pogoste napake. Mejniki razvoja IT so torej:

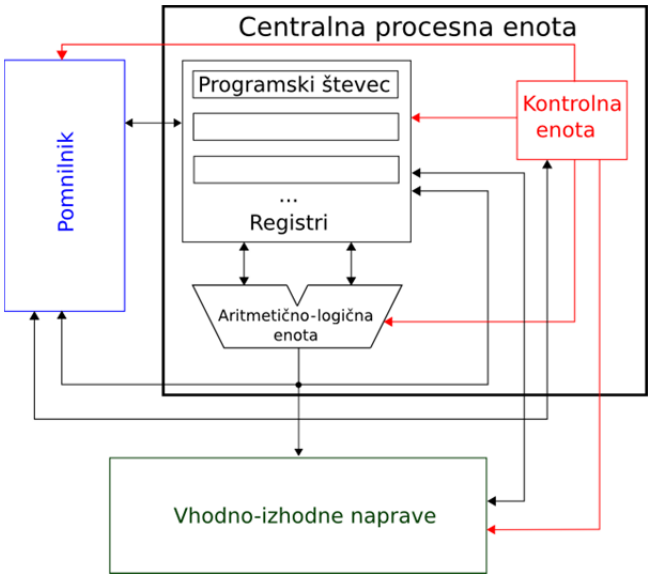
- 1. generacija (1940–56): **elektronke**, ročni vnos podatkov, velika poraba energije, npr. ENIAC iz leta 1946;
- 2. generacija (1956–63): **tranzistorji**, s simbolnim jezikom so poenostavili vnašanje in branje podatkov;
- 3. generacija (60. leta): **integrirana vezja** = **čipi**, večja hitrost, zaporedni ukazi in opisni programski jezik;
- 4. generacija (1971–): **mikroprocesor**, več elementov, npr. Intel 8086, IBM PC;
- 5. generacija (2010–): **vzporedno procesiranje**, visoka integracija, umetna inteligenca.

Prekomerna in neustrezna uporaba informacijske tehnologije vpliva na **zdravje** ljudi (npr. dolgotrajno sedenje, hrup, suh zrak itd.). **Ergonomija** opredeljuje ustrezno organizacijo in ureditev delovnega mesta. Delovno mesto naj bo prostorno, zračno, svetlo in udobno. Prilagodimo zaslon, sedežni položaj, redno se razgibavamo in zračimo.

Zgradba in delovanje računalnika

Von Neumannov model (arhitekturo) računalnika sestavljajo trije deli:

- **centralna procesna enota** (**CPE**),
- **pomnilnik**,
- **vhodno-izhodne naprave**.



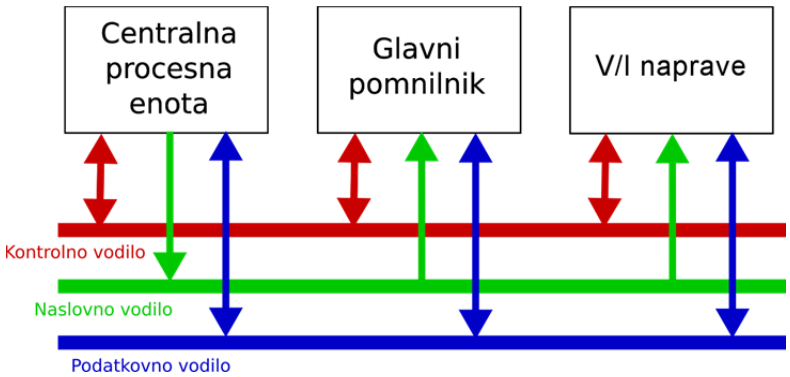
Osrednjo vlogo ima CPE, ki program izvaja tako, da **ukaze zaporedno prevzema iz pomnilnika** ter jih **izvršuje**. Ukaz opravi operacijo na podatkih, ki jim pravimo **operandi**. Operandi so lahko shranjeni v pomnilniku ali pa so prebrani iz vhodno-izhodnih naprav.

Vhodno-izhodne naprave omogočajo **komunikacijo CPE z okolico** (to so npr. tipkovnica, zaslon, grafična kartica itd.).

Centralna procesna enota (**CPE**) je sestavljena iz:

- **registrov**, v katerih hrani operande, s katerimi računa,
- **aritmetično-logične enote** (**ALE**), ki izvaja aritmetične in logične operacije,
- **kontrolne enote**, ki nadzira delovanje CPE.

Kontrolna enota prebere ukaz iz glavnega pomnilnika, ki se nahaja na naslovu, ki ga hrani **programski števec** (**PC**). Programski števec je eden od registrov v CPE; ima zelo pomembno vlogo, saj hrani naslov naslednjega ukaza, ki naj se izvede. Ko ukaz izvrši, se programski števec poveča za 1: PC = PC + 1.



Sodobni računalniki uporabljajo razširjeno von Neumannovo arhitekturo, ki vsebuje **sistemsko** (oz. **centralno**) **vodilo**, ki povezuje CPE, pomnilnik in vhodno-izhodne naprave.
Preko **kontrolnega vodila** CPE komunicira z drugimi napravami in prejema povratne signale.
Preko **podatkovnega vodila** se vrši prenos podatkov med CPE in pomnilnikom.
Preko **naslovnega vodila** se izvaja direktno naslavljanje pomnilnika (naslov operandov), širina vodila pa je odvisna od CPE (npr. 64-bitna).

Pomnilniki so naprave, ki hranijo informacije. Delimo ga na:

- **glavni** pomnilnik (primarni, delovni pomnilnik),
- **pomožni** pomnilnik (sekundarni, zunanji pomnilnik).

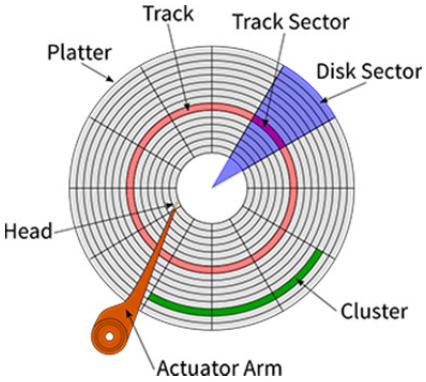
Glavni pomnilnik je linearno zaporedje **pomnilniških besed**, v katere zapisujemo in iz katerih beremo vsebino. Pomnilniška beseda je danes največkrat **8 bitov** (1 bajt), vsak bit pa predstavlja eno pomnilniško celico. Vsaka pomnilniška beseda ima svoj **enoličen naslov**. Branju in zapisovanju v pomnilnik pa pravimo **pomnilniški dostop**. V pomnilnik ne moremo vpisati oz. prebrati manj kot ene pomnilniške besede (ne moremo brati posameznih bitov).
Danes uporabljamo predvsem **pomnilnik z naključnim dostopom** (**RAM**). Dve glavni vrsti:

- **dinamični RAM** (**DRAM**): za en bit 1 tranzistor, vsebina se stalno osvežuje,
- **statični RAM** (**SRAM**): za en bit 6 tranzistorjev, osveževanje ni potrebno.

Statični RAM je hitrejši od dinamičnega (ni osveževanja), je pa dražji. Oba ne omogočata trajnega hranjenja podatkov, ob odklopu napajanja se podatki izgubijo.

Pomožni pomnilnik trajno shranjuje podatke, poznamo več vrst:

- **bralni pomnilniki** (**ROM**): pisanje ni mogoče ali je zelo zamudno, sem spadajo USB, SSD, kartice SD, CD, DVD,
- **trdi diski** (**HDD**): diski, prevlečeni z magnetno plastjo, podatki so zapisani v koncentričnih krogih; bralno-pisalna glava preskakuje med sledmi.



Trdi disk se **formatira** z obdelavo magnetne snovi: dobimo množico magnetkov, ki vsak predstavlja po 1 bit. Disk je razdeljen na **particije**, ki so lahko:

- **primarne**: OS se naloži iz te particije, lahko jih je največ 4,
- **razširjene**: razdeljena je na logične pogone (C:, D: itd.), lahko je največ 1.

Primarna particija ima lahko le en **datotečni sistem**, razširjena pa jih lahko ima več glede na število logičnih pogonov. Najpogostejši so NTFS, exFAT, FAT32 itd.
Disk je sestavljen iz **zaporedja blokov** (najmanjših enot za pisanje ali branje), vsaka datoteka je zapisana v enem ali več blokih (zato ostane nekaj neizkoriščenega prostora). Do **fragmentacije** pride, ko disk datoteke ne zapiše v zaporedne, temveč v razpršene bloke (zaradi npr. pogostega brisanja, pisanja). Disk shranjuje **tabelo prostih blokov**. **Mapa** je praktično imenik za lokacije datotek oz. blokov (datoteka, ki vsebuje naslove datotek).

Pomnilniki so urejeni v **pomnilniško hierarhijo**:

- CPE in predpomnilnik L1 → L2, L3 cache → RAM → SSD/HDD

Dostope, pri katerih je potreben prenos podatkov s počasnejših nivojev hierarhije, imenujemo **zgrešitve** v predpomnilniku. **Lokalnost pomnilniških dostopov** je lastnost von Neumannovih računalnikov, ki pogosto dostopajo do naslovov v pomnilniku, ki so si med seboj blizu in se v kratkem časovnem intervalu pojavljajo večkrat. **Navidezni pomnilnik** je del RAM-a na disku v swap datoteki.

Delovanje računalnika je določeno z zaporedjem ukazov: **ukazi** hranijo informacijo o **operaciji**, ki naj jo izvede procesor, in o morebitnih **operandih** operacije. **Operandi** so lahko shranjeni v registrih, glavnem pomnilniku, vhodno-izhodnih napravah ali pa so vsebovani že v samem ukazu.

Operacijska koda	Informacija o operandih

Format ukaza je razdelitev bitov na posamezna polja, ki hranijo informacijo o operaciji in operandih. Poznamo več vrst ukazov: **aritmetično-logični** (ADD), **prenos podatkov** (LOAD, STORE) in **skoke** (JUMP). Glede na **način naslavljanja** operandov ločimo:

- **takojšnje**: vrednost operanda je podana v ukazu,
- **vsebovano**: ni treba povedati, kje je operand (npr. če imamo samo 1 register),
- **direktno**: povemo, kje je operand (št. registra/naslov v RAM),
- **indirektno**: podamo naslov naslova operanda.

Strojna oprema računalnika

Računalnik je sestavljen iz **procesorja**, **hladilnika za procesor**, **matične plošče**, pomnilnika **RAM**, pomožnega pomnilnika (**HDD**, **SSD**), **grafične kartice**, **ohišja**, **napajalnika** in drugih **vhodno-izhodnih naprav**. To so npr. miška, tipkovnica, zaslon itd. Seveda moramo paziti na **združljivost** (*kompatibilnost*) naprav – matična plošča mora imeti ustrezno podnožje (*socket*) za CPE, RAM mora ustrezati matični plošči (npr. DDR4) itd. Pri izbiri komponent pazimo na hitrost CPE/GPE/RAM, število jeder, količino pomnilnika …

Programska oprema računalnika

Programsko opremo delimo na:

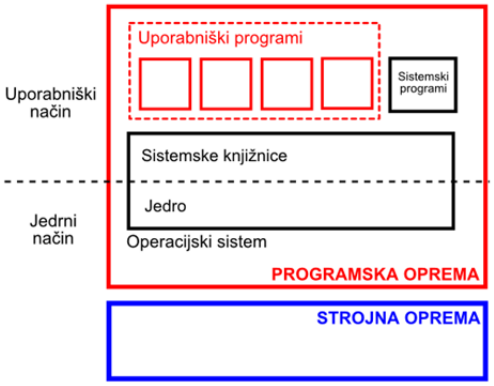
- **sistemsko**: operacijski in datotečni sistem,
- **uporabniško**: urejevalniki teksta, slikarski programi …

Sistemska orodja so programi, ki skrbijo za optimalno izkoriščanje pomnilnika, diska, stiskanje podatkov ipd. **Gonilniki** skrbijo za ustrezno delovanje priključenih I/O naprav.

Operacijski sistem je posrednik med programsko in strojno opremo:

- **jedro** nadzira računalnik in zagotavlja zanesljivo delovanje,
- **uporabniški vmesnik** jedro približa uporabniku (grafični vmesnik/ukazna vrstica).

Glede na **število uporabnikov** ločimo eno- in večuporabniške OS.
Glede na **programe** ločimo eno- in večopravilne OS.



Programsko opremo ponavadi samo uporabljamo in si je ne lastimo, temu pravimo **licenca**. Glede na **licenco** ločimo več vrst programske opreme:

- **javna** (*public domain*): brezplačna, uporabljamo brez omejitev,
- **prosta** (*free software*): uporabljamo jo za katerikoli namen in z njo razpolagamo,
- **odprtokodna** (*open source*): tako kot prosta, dovoljuje še dostop do izvorne kode,
- **poskusna** (*trial*): brezplačna, delovanje je omejeno na nekaj dni,
- **tržna** (*commercial software*): uporablja se v tržne namene (za zaslužek).

Računalniški vsiljivci so programi, katerih izključni namen je povzročanje škode. To so:

- **trojanski konj**: predstavlja se kot drug program,
- **virus**: okuži nek program, uničuje podatke, izpisuje na zaslon,
- **bomba**: čaka na izpolnitev pogojev, nato napade,
- **zajček**: množi se nenadzorovano, obremenjuje CPE, pomnilnik in disk,
- **črv**: tako kot zajček, okuži pa tudi računalnike v omrežju,
- **spam**: nadležno sporočilo z namenom vsiljevanja vsebine,
- **potegavščina**: namerno širi neresnice, straši uporabnike.

Najboljša zaščita pred računalniškimi vsiljivci je dobro nastavljen **požarni zid** (*firewall*), **antivirusni programi**, redno posodabljanje OS, konec koncev pa zdrava kmečka pamet – ne odpiramo sumljivih datotek, uporabljamo zahtevna gesla ipd.

Računalniška omrežja

Računalniško omrežje je množica med seboj povezanih avtonomnih naprav. Sestavljajo ga:

- **uporabniški računalniki** (*hosts*),
- **vozliščni računalniki**,
- **prenosne linije**.

Povezovanje računalnikov v omrežje je pomembno zaradi:

- sodobnega komuniciranja in enostavnejšega dostopa do informacij,
- prenosa podatkov med računalniki,
- deljenja virov,
- prihranka denarja in skupne rabe drage opreme.

Glede na vrsto poznamo **lokalna omrežja** (**LAN**) in **omrežja širokega dosega** (**WAN**).

Protokol je predpisan način, skupek pravil, s katerim si dva ali več računalnikov ali drugih omrežnih naprav izmenjuje podatke. V internetu je uporabljen protokol TCP/IP.

Podatki po omrežju potujejo v **paketkih**. Temu pravimo **paketni prenos podatkov**, pri katerem ni potrebno vzpostaviti sočasne povezave med izvorom in prejemnikom:

- paketnik (PAD) podatke sestavlja v pakete in jih razstavlja,
- paketi se v vozliščih usmerjajo k naslovniku, lahko potujejo po različnih poteh,
- prispeti paketi se razvrstijo v pravilni vrstni red.

Internet je *omrežje omrežij*, ki deluje na protokolu TCP/IP, naprave pa se prepoznavajo po IP številkah. Ogrodje interneta tvorijo internetni strežniki (DNS). TCP opisuje vsebino paketkov in skrbi za njihovo berljivost, IP pa je metoda za prenos informacij med računalniki.

Primeri protokolov

SMTP	pošiljanje e-pošte
POP3, IMAP	branje e-pošte
PPP	povezava v internet preko modema
http, https	delo v svetovnem spletu
ftp, sftp	prenos datotek
telnet, ssh	delo na oddaljenem računalniku
ntp	uskladitev časa na računalniku

Vsaka naprava v omrežju TCP/IP ima svoj **IP naslov**. IP številka določa položaj računalnika v svetovnem omrežju. Poznamo dva standarda:

- **IPv4**: 32-bitna št. (štiri 8-bitna št., ločena s piko), vsako št. ima lahko vrednost od 0 do 255, npr. `193.2.1.66`,
- **IPv6**: 128-bitna št. (osem šestnajstiških št., ločenih z dvopičji), niz ničel lahko enkrat nadomestimo z dvema podpičjema, npr. `AAAA:0:0:0:BBBB:0:0:CCCC` = `AAAA::BBBB:0:0:CCCC`

IPv4 naslove lahko pretvorimo v IPv6 naslove, obratno pa ne. Primer:

IPv4 naslov = `192.1.2.51`

IPv6 naslov = `0:0:0:0:0:0:192.1.2.51` oz. `::192.1.2.51`

Poleg lokalnih in omrežij širokega dosega ločimo še **omrežja enakovrednih računalnikov** (*peer-to-peer*) in omrežja tipa **uporabnik/strežnik** (najpogostejša). Pri peer-to-peer omrežjih uporabnik neposredno komunicira s katerimkoli drugim računalnikom v omrežju.
Pri omrežjih tipa uporabnik/strežnik imamo:

- **strežnik**: eden ali več velikih osrednjih računalnikov, ki hranijo podatke in programe,
- **delovne postaje** (**uporabniki**): podrejene, komunicirajo le s strežnikom.

Domenski strežnik (**DNS**) spreminja IP naslove v imena in obratno. Pozna vse IP naslove in imena v omrežju pod sabo in zna spraševati tiste nad sabo.

Topologija omrežja je način, kako so omrežne naprave fizično povezane med seboj:

- **vsak z vsakim**: samo pri manjših omrežjih,
- **vodilo**: naprave si delijo skupno povezavo, okvara vodila pa onemogoči promet,
- **obroč**: sklenjena povezava kot vodilo, okvara tudi onemogoči promet,
- **zvezda**: centralno vozlišče (stikalo), do naprav pa je vzpostavljena povezava vsak z vsakim, okvara vodila onemogoči promet.

Omrežne naprave omogočajo povezavo med omrežji. Poznamo:

- **hub** (razdelilnik): podatke pošilja vsem, ki so priključeni,
- **switch** (stikalo): podatke pošilja le napravi, kateri so namenjeni; naučijo se, kaj je priključeno na določenih vratih glede na MAC naslove – tabela naslovov CAM,
- **router** (usmerjevalnik): povezuje dve omrežji (lokalna in prostrana); filtrira podatkovne pakete in jih prepušča le v segmente, v katere so namenjeni; z DHCP tudi dodeljuje IP naslove; deluje kot požarni zid (*firewall*).

Elementi LAN omrežja so:

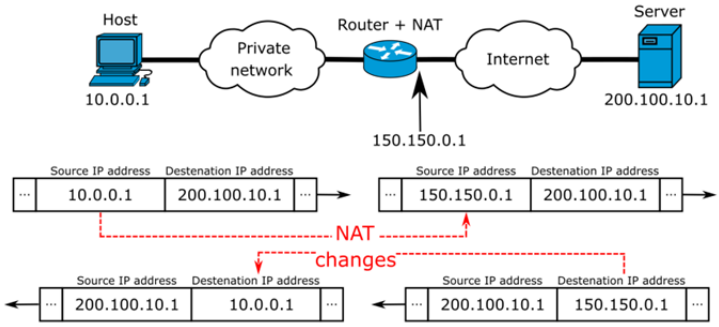
- **omrežna kartica** z MAC naslovom (48-bitni, 6 šestnajstiških števil),
- **omrežni kabel** (UTP) s posukanim parom za odpravo motenj.

ARP je protokol, ki IP naslove povezuje z MAC naslovi. Usmerjevalnik (router) v omrežju »vpraša«, katera naprava ima IP naslov npr. `192.168.1.10`; javi se naprava in usmerjevalniku pove svoj MAC naslov (npr. `00:50:56:88:13:bd`).

Če naprava ugotovi, da je ciljni naslov izven omrežja, naredi ARP poizvedbo po IP naslovu **privzetega prehoda** (*default gateway*) – to je **IP naslov usmerjevalnika**. Usmerjevalnik odgovori s prehodom v drugo omrežje in svojim MAC naslovom. Naprava usmerjevalniku pošlje paketek in preveri ciljni naslov. Usmerjevalnik naredi ARP poizvedbo v drugo omrežje in pošlje paketek na nov ciljni MAC naslov.

Nastavitev omrežja:

- **NAT** (*network address translation*): notranje IP naslove naprav v paketkih prevaja v zunanji IP naslov omrežne naprave, ki predstavlja lokalno omrežje in jih nato posreduje naprej,



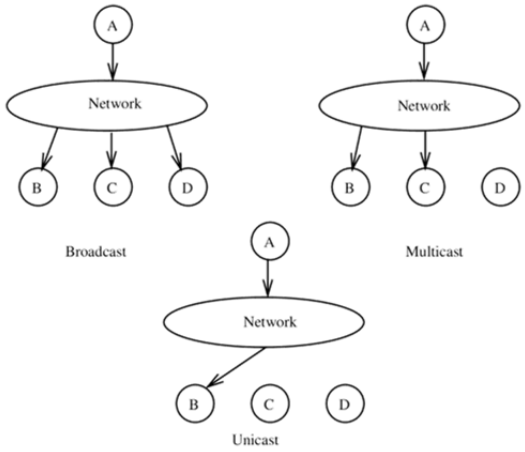
- **fiksni/dinamični IP**: fiksni so dobri za strežnike, dinamični pa za vsakdanje uporabnike,
- **privzeti prehod**: je IP naslov usmerjevalnika, preko katerega prehajajo paketki,
- **maska podomrežja**: določa velikost naslovnega prostora lokalnega omrežja, 255 določa naslov omrežja, 0 pa naslovni prostor za naprave; npr. `255.255.255.0` pomeni, da imajo naprave lahko IP naslove od `192.168.1.0` do `192.168.1.255`,
- **vrata** oz. **porti**: delujejo kot interne telefonske številke,
- **številka omrežja**: npr. `193.2.145.128`.

Model TCP/IP ima pet **plasti**:

1) **aplikacijska**: pošilja in prevzema toke podatkov od oz. do transportne plasti,
2) **transportna**: vzpostavlja povezavo med procesi v računalnikih,
3) **omrežna**: ustvarja paketke, skrbi za iskanje poti paketkov v omrežju,
4) **povezavna**: skrbi za prenos med neposredno povezanima napravama,
5) **fizična**: pretvori ničle in enice v signal (frekvence, konektorji, kabli).

Načini posredovanja sporočil so:

- **unicast**: ena naprava pošlje drugi, point to point,
- **broadcast**: ena naprava pošlje vsem, uporablja se pri ARP,
- **multicast**: ena naprava pošlje več določenim, a ne vsem.



Bluetooth povezuje do 8 naprav na razdalji 10 metrov. Uporablja se za brezžični prenos podatkov med napravami v omrežju zelo omejenega dosega.

Najpomebnejše internetne storitve so:

- **splet**: www strežniki, uporablja se protokol http oz. https, HTML in CSS,
- **e-pošta**: urejanje, pošiljanje, sprejemanje in branje elektronskih sporočil,
- **telnet**, **ssh** in **ftp**, **sftp**.

Internetni odjemalec oz. **spletni brskalnik** pozna vsaj http protokol, komunicira s strežniki in prikazuje podatke, ki jih pošilja in sprejema. Npr. Chrome, Firefox, Opera, Safari, Edge.

E-pošta uporablja hudo star protokol SMTP za pošiljanje (tekst je v ASCII), vendar se v današnjem času protokol zaradi varnosti nadgrajuje z enkripcijo; za prejemanje e-pošte se uporabljata protokola POP3 in IMAP.

VARNOST V OMREŽJIH

Ko govorimo o računalniški varnosti, ponavadi upoštevamo dve lastnosti:

- **zaupnost**: sporočil ne more prebrati vsakdo, ampak samo oseba, ki so ji namenjeni,
- **zanesljivost**: sistem ali aplikacijo lahko uporabljamo, kadarkoli jo potrebujemo.

Nevarnosti v omrežjih so **onemogočanje storitve** (DoS/DDoS), **kraja osebnih podatkov** ali **identitete**, **vdiranje v računalnik**, **man-in-the-middle** (npr. napadalec prepriča žrtev, da je privzeti prehod, ves promet v internet gre skozi napadalčev računalnik – ARP spoofing, lažni DHCP strežniki, zastrupljanje predpomnilnika DNS) ipd.

Varnostni mehanizmi so: (IAAA - identification, authentication, authorisation, accounting)

- **identifikacija**: predstavitev uporabnika (npr. vnos uporabniškega imena),
- **avtentikacija**: preverjanje pristnosti uporabnika (npr. vnos gesla),
- **avtorizacija**: ugotavljanje pravic uporabnika v sistemu (npr. vstop v svoj nabiralnik),
- **beleženje**: pregledovanje zabeležene zgodovine aktivnosti (npr. napadalec skuša ugotoviti naše geslo, po nekaj poskusih mu sistem onemogoči prijavo).

Zavarujemo se lahko z **dobrim geslom**, **dvostopenjsko avtentikacijo** (pametna kartica, telefon) ali celo **biološkim preverjanjem** (prstni odpis, sken roženice).

Osnova za vse moderne varnostne mehanizme so **neobrnljive funkcije**, funkcije brez inverzne funkcije, kot je npr. `y = (7 * x) % 3`. Seveda lahko z brute-force napadalci ugotovijo rezultate majhnih števil, vendar naši računalniki uporabljajo več kot desetmestna števila, zato je natanko določeno število zelo težko najti.

Zgoščevalne funkcije (*hashing functions*) različno dolgim vrednostim `b` priredijo zgoščene vrednosti `p` fiksne dolžine. Primer zgoščevalne funkcije je MD5 ali SHA-256 ipd. Iz `p` je praktično nemogoče izračunati `b`, $\texttt{hash}(b_1) = \texttt{hash}(b_2)$ pa je skoraj neverjetno. Gesla zato ponavadi shranjujemo kot **soljene zgoščene vrednosti**. Geslu pripnemo sol in vse skupaj nato zgostimo (*hashiramo*), zato je napadalcu iz baze zelo težko ugotoviti gesla. Preverjanje gesla je lahko, ker geslo preverimo, sol pa je shranjena v bazi.

```
geslo = banana
sol = 29323adesg8394mn
→ izračunamo hash od gesla in soli skupaj >>> hash(geslo + sol)
```

KRIPTOGRAFIJA

Pri **simetrični kriptografiji** imata oba uporabnika enak ključ. Primer simetrične kriptografije je Cezarjeva ali Viegenerjeva šifra. Pri Cezarjevi šifri abecedo zamaknemo za nekaj mest, ključ pa je zamik. Viegenerjeva šifra deluje na podoben način, le da je vsaka črka zamaknjena glede na ključ. Če je ključ pri Viegenerjevi šifri popolnoma naključen in enako dolg ali daljši od sporočila, je šifra načeloma varna. Glej Wiki za razlago. Problem pri simetrični kriptografiji je varna izmenjava ključev.

Pri **asimetrični kriptografiji** se kot osnova uporablja Diffie-Hellmanova izmenjava ključev. Vsak ima par ključev (`J` - javni, `P` - privatni) in sporočilo `s`. Velja `J(P(s)) = P(J(s))`.

```
J_B(s) → A pošlje sporočilo, ki ga lahko prebere le B
J_B(P_A(s)) ali J_B(s + P_A(podpis)) → sporočilo z digitalnim podpisom
J_B(s + P_A(hash(s))) → celovito sporočilo
```

Overitelj (CA = Certificate Authority) jamči, da par ključev res pripada uporabniku – overitelju vsi uporabniki zaupamo. V praksi si asimetrično izmenjamo le ključe, kriptiramo pa simetrično, ker je veliko hitrejše.