**Zapiski**

1. **Pretvorba med številskimi sistemi**
   1. Desetiško v dvojiško (primer 125(10))

125 % 2 = 62 + 1

62 % 2 = 31 + 0

31 % 2 = 15 + 1

15 % 2 = 7 + 1

7 % 2 = 3 + 1

3 % 2 = 1 + 1

1 % 2 = 0 + 1

Preberemo v obratnem vrstnem redu 🡪 1111101(2).

Za 8 bitno pretvorbo dodajamo spredaj 0 dokler nimamo skupaj 8 števk 🡪 01111101(2).

* 1. Dvojiško v desetiško (primer 01111101(2))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Seštejemo vrednosti spodnje vrstice tam kjer je binarna vrednost 1 🡪 1+4+8+16+32+64=125(10)

* 1. Desetiško v druge številske sisteme (primer 125(10) v šestnajstiško)

125 % 16 = 7 + 13 (D)

7 % 16 = 0 + 7

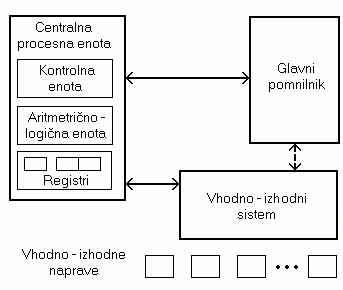
Preberemo v obratnem vrstnem redu 🡪 7D(16).

Za pretvorbo v ostale številske sisteme delimo s primernim številom in vzamemo ostanek.

* 1. Šestnajstiško v dvojiško (primer 7D(16))

7D(16)=0111 1110(2) 🡪 pretvorimo v desetiško po algoritmu za pretvorbo iz dvojiškega v desetiški sistem.

**Von Neumannov model računalnika**



* Računalnik mora biti obče uporaben in mora izvajati program avtomatsko.
* Ukazi naj bodo shranjeni v enaki obliki in v isti enoti računalnika (pomnilniku) kot podatki, ki jih obdeluje.
* Računalnik mora vsebovati centralno procesno enoto (CPE) in vhodno-izhodne enote. CPE mora biti zgrajena iz:
  + aritmetično-logične enote, ki bo izvajala operacije računanja,
  + krmilne enote, ki bo razumela ukaze iz pomnilnika in upravljala z delovanjem računalnika,
  + registrov:
    - akumulator (shranjevanje operandov),
    - programski števec (kaže na naslov naslednjega ukaza),
    - indeksni register (pri indeksnemu naslavljanju),
    - skladovni kazalec (kaže na vrh sklada),
    - statusni register,
* Preko vhodno-izhodne enote bo računalnik izmenjeval podatke med uporabnikom in okolico.
* Delovanje računalnika naj bo zasnovano na osnovi dvojiškega številskega sestava. To je potrebno zaradi električne realizacije računalnika. Električna vezja najlažje ločijo dve diskretni vrednosti. Drugi razlog je logične narave, saj bo računalnik namenjen reševanju logičnih problemov, opisanih s stanjema da ali ne.
* Von Neumannov model računalnika podpira ukaze:
  + pridobitev ukaza (fetch)
  + njegovo dekodiranje (decode)
  + izvedba ukaza (execute)
  + zapis rezultata (writeback).

1. **Ukazi v operacijskih sistemih**

|  |
| --- |
| **Ukaz za izpis vsebine imenika:**  DOS: dir  Linux: dir [opcije] imenik  Opcije:  -a --all (izpiše vse datoteke in imenike, tudi skrite),  --color (izpiše datoteke in imenike v različnih barvah) |

|  |
| --- |
| **Ukaz za spremembo delovnega imenika:**  DOS: cd cd Desktop, cd...  Linux: cd [opcije] imenik cd ~/Desktop  Opcije:  -a --all (izpiše vse datoteke in imenike, tudi skrite),  --color (izpiše datoteke in imenike v različnih barvah) |

|  |
| --- |
| **Ukaz za izpis trenutnega imenika:**  DOS: chdir  Linux: pwd |

|  |
| --- |
| **Kopiranje datotek in imenikov:**  DOS: copy cilj copy Slika1.jpg C:\Desktop  Linux: cp [opcije] izvor cilj cp Slika1.jpg ~/Desktop  Opcije:  -r (prekopira imenike),  -f ali --force (prepiše obstoječe cilje, brez opozorila),  -i ali --interactive (vpraša, preden prepiše obstoječe datoteke),  -v (izpiše dodatne informacije med postopkom) |

|  |
| --- |
| **Ukaz za premikanje datotek in imenikov:**  DOS: move cilj move Slika1.jpg C:\Desktop  Linux: mv [opcije] cilj mv Slika1.jpg ~/Desktop  Opcije:  -r (prekopira imenike),  -f ali --force (prepiše obstoječe cilje, brez opozorila),  -i ali --interactive (vpraša, preden prepiše obstoječe datoteke),  -v (izpiše dodatne informacije med postopkom) |

|  |
| --- |
| **Ukaz za brisanje datotek ali imenikov:**  DOS: del del Slika1.jpg  Linux: rm [opcije] datoteka rm Slika1.jpg  Opcije:  -r (prekopira imenike),  -f ali --force (prepiše obstoječe cilje, brez opozorila),  -i ali --interactive (vpraša, preden prepiše obstoječe datoteke),  -v (izpiše dodatne informacije med postopkom) |

|  |
| --- |
| **Ukaz za ustvarjanje imenikov:**  DOS: md  Linux: mkdir |

|  |
| --- |
| **Kreiranje datoteke:**  DOS: copy con copy con a.txt (besedilo + F6) Linux: cat [datoteka] cat a.txt (besedilo, CTR + D) |

|  |
| --- |
| **Iskanje vsebine po datotekah:**  DOS: find >>string<< [datoteka] find »text« C:\Desktop\a.txt Linux: grep [opcije] datoteka grep »text« a.txt  Opcije:  -i (zanemari CapsLock),  -w (išče samo besede),  -c (šteje vrstice ujemanj). |

|  |
| --- |
| **Urejanje vsebine po datotekah:**  DOS: edit [datoteka] edit text.txt  Linux: vi [datoteka] vi text.txt (na koncu ESC in ZZ) |

|  |
| --- |
| **Zaustavitev sistema:**  DOS: shutdown [opcija]  /s – zaustavljanje sistema, /r – ponovni zagon, /a – preklic  Linux: shutdown [opcija]  Opcije:  -h (zaustavitev),  -r (ponovni zagon),  -c (preklic zaustavitve)  Čas: podamo čas ali napišemo now  Sporočilo: sporočilo, ki je poslano vsem uporabnikom pred izklopom. |

|  |
| --- |
| **Sprememba lastništva datotek in imenikov:**  Linux: chmod vzorec datoteka  Ukaz chmod se uporablja za nastavljanje oz. spreminjanje dovoljenj datotek. Novo nastalim datotekam je navadno podeljena vrednost -rw--r--w--, ki omogoča ogled vsem uporabnikom sistema. Dovoljenja lahko spreminja le lastnik datoteke ali root.  To lahko storimo v dveh oblikah:   * Simbolni:   chmod {a,u,g,v}{+,-}{r,w,x} imeDatoteke  Simbolna oblika je sestavljena iz ene ali več črk »a«, »u«, »g« ali »v«.  Skupine, katerim določimo dovoljenja so:   * + »a« (vsi – all),   + »u« (uporabnik ali lastnik – user),   + »g« (skupina – group) in   + »o« (ostali – other).   Dodajanje oz. odvzemanje pravic:   * + »+« (pravice se dodajo),   + »-« (pravice se odvzamejo).   Vrste dovoljenj datotek:   * + »r« (branje – read: uporabnikom dopušča branje vsebine datoteke ali pri imenikih izpisati vsebino imenika),   + »w« (pisanje – write: uporabnikom dopušča spreminjati vsebino datotek ali imenikov),   + »x« (izvajanje – executable: uporabnikom dopušča pogon datoteke. Pri imenikih omogoča premik v ta imenik). * Ali kot osmiško število   chmod številčnaVrednostDovoljenj imeDatoteke  Vrednost številk ponazorimo skupin po vrsti združimo in dobimo trimestno osmiško število. Osmiško število dobimo tako, da seštejemo vrednosti dovoljenj:  Dovoljenje za branje ima vrednost 4, za pisanje 2, za izvajanja pa 1. Tako bi 711 pomenilo, da ima lastnik vsa dovoljenja, ostali pa le dovoljenje za izvajanje.  -UGO |

**Operacije nad procesi**

|  |
| --- |
| **Izpis seznama nad procesi:**  Linux: ps [opcije]  Opcije:  -i (izpiše informacije o roditeljih procesa in prioriteta, s katero se proces poganja),  -a (izpiše vse procese, tudi drugih uporabnikov),  -r (izpiše delujoče procese). |

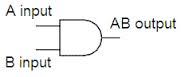
|  |
| --- |
| **Izpis procesov po uporabi procesorja:**  Linux: top [opcije]  Opcije:  -d čas (določimo interval osveževanja)  Ukaz top nam v realnem času prikazuje seznam najbolj aktivnih procesov. Za vsak proces podaja tudi odstotek uporabljenega pomnilnika in procesorskega časa. |
| **Izpis procesov po uporabi procesorja:**  Linux: renice prioriteta [-p pid]  Prioriteta: navedemo številko, za koliko hočemo povečati oz. zmanjšati prioriteto procesa. Prioriteta lahko zavzema vrednosti od -20 do 20, pri čemer nam -20 pomeni največje obremenjevanje sistema, vrednost 20 pa pomeni, da bo ukaz dobil najmanj procesorskega časa. | |

|  |
| --- |
| **Končevanje procesov:**  Linux: kill [signal] pid  Signali:  1 sighup (uporablja se za vnovično nastavitev procesov  2 sigint (uporablja se za zaustavitev med izvajanjem)  9 sigkill (brezpogojno končevanje – ni mogoče prekiniti)  15 sigterm (privzeta vrednost, uporablja se za milo zaustavitev procesov) |

|  |
| --- |
| **Izpis informacije o sistemu:**  Linux: uname [opcije]  Opcije:  -a (izpiše vse informacije)  -m (izpiše tip računalnika)  -n (izpiše ime računalnika)  -r (izpiše verzijo sistema)  -s (izpiše ime sistema)  -p (izpiše tip procesorja) |

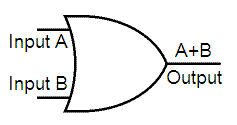
1. **Boolova algebra**
   1. Logični IN

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



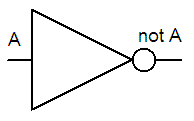
* 1. Logični ALI

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



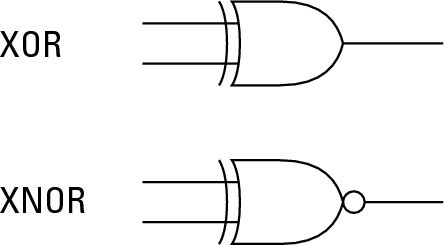
* 1. Logični NE

|  |  |
| --- | --- |
| 0 | 1 |
| 0 | 0 |

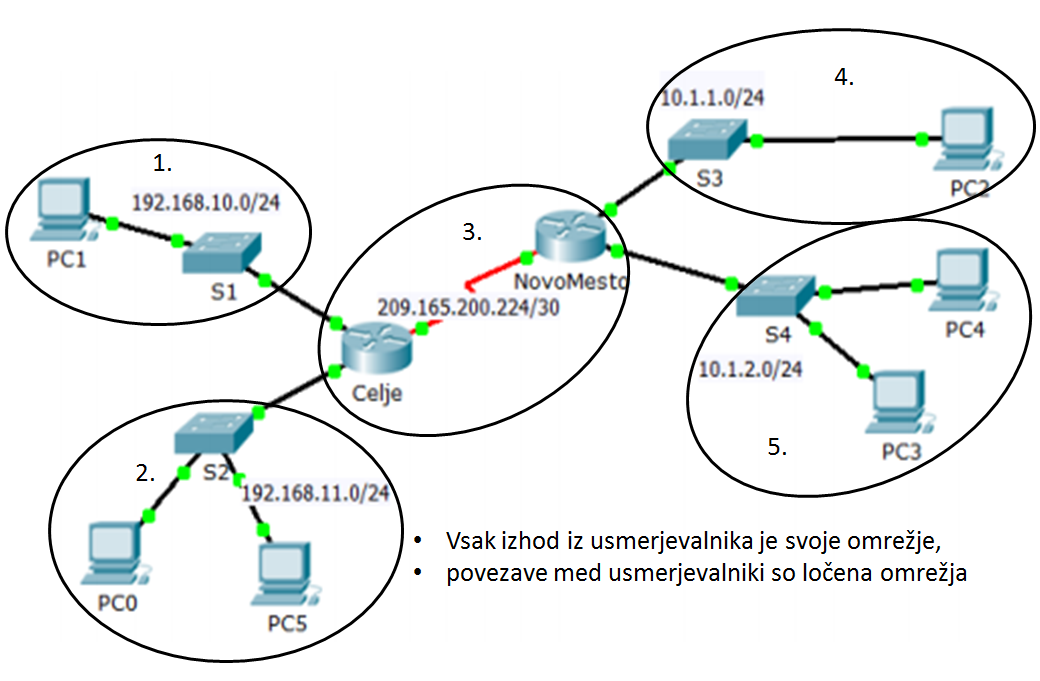


* 1. Ekskluzivni ALI

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



1. **Omrežja**



**VLSM**

|  |  |  |  |
| --- | --- | --- | --- |
| 0  127 | 128  191 | | |
| 192  223 | 224  239 | |
| 240  247 | 248  251 |
| 252  255 |

Prvi naslov je naslov omrežja, zadnji naslov je broadcast naslov.

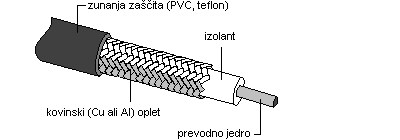
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IP naslovi | | |  |  |
| razred | začetni naslov | končni naslov | maska | št. naslovov |
| A | 1.0.0.0 | 126.255.255.255 | /8 do /15 | 16M |
| B | 127.0.0.0 | 191.255.255.255 | /16 do /23 | 64.000 |
| C | 192.0.0.0 | 223.255.255.255 | /24 do /30 | 254 |

**Primerjava OSI in TCP/IP modelov:**

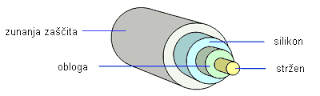
|  |  |  |
| --- | --- | --- |
| **7.** | Aplikacijska plast | Aplikacijska plast |
| **6.** | Predstavitvena plast |
| **5.** | Plast seje |
| **4.** | Prenosna plast | Prenosna plast |
| **3.** | Mrežna plast | Omrežna plast |
| **2.** | Povezavna plast | Dostopna plast |
| **1.** | Fizična plast |

|  |
| --- |
| **Računanje maske omrežja**  255.255.255.m  k.. število podomrežij |

|  |  |
| --- | --- |
| Vrsta kabla | Lastnosti |
| koaksialen kabel | malo občutljivi na EM, poceni, upogljivi |
| sukana parica (UTP) | malo občutljivi na EM, poceni |
| brezžični medij | občutljivi na EM… upogibej jih kokr čš |
| optični medij | neobčutljivi na EM, dragi, neupogljivi |

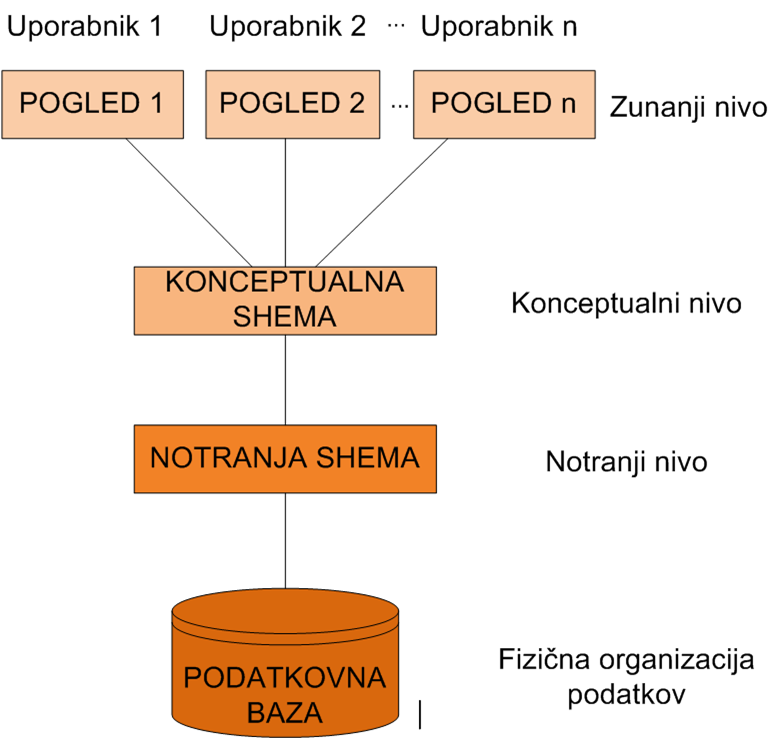


Slika 1: koaksialen kabel



Slika 2: optični kabel

1. **Podatkovne baze in informacijski sistemi**
   1. ANSI/SPARC



* 1. Redundanca podatkov:
* v neki bazi podatki shranjeni večkrat,
* poveča potrebo po vnosu in posodabljanju podatkov,
* vpliva na skladnost podatkovne baze.
  1. USE-CASE diagramom opišemo statično strukturo sistema. Uporablja se za ustvarjanje in risanje diagramov, ne pa za normalizacijo podatkovne baze. Na kontekstnem nivoju je lahko samo en proces
  2. CLASS diagram opišemo objekte, atribute in asociacije.
  3. Vrste informacijskih sistemov:
* sistem za upravljanje delovnih procesov,
* transakcijski informacijski sistemi,
* upravno-ravnateljaljevalni informacijski sistem,
* odločitveni sistemi,
* ekspertni sistemi
  1. Diagram toka podatkov

Gradniki so: vsi terminatorji (zunanje entitete), tokovi med terminatorji in sistemskim procesom ter sistemski proces. Za opisovanje toka podatkov uporabljamo podatkovne slovarje.