

Ključne karakteristike memorijskih sistema

Ključne karakteristike memorijskih sistema

- 1. Položaj memorije:
- prema tome da li je memorija unutar ili van računara podela je na:
 - **Unutrašnja memorija:** glavna memorija, registri procesora, keš memorija
 - **Spoljašnja memorija-** su periferni uređaji za skladištenje podataka kojima se pristupa putem U/I kontrolera: optički diskovi, magnetni diskovi, magnetne trake

Ključne karakteristike memorijskih sistema

- 2. Kapacitet memorije

- Za unutrašnju memoriju tipično se iskazuje kao broj reči (8, 16, 32 bita) ili broj bajtova
- Za spoljašnju memoriju tipično se iskazuje u bajtovima

Ključne karakteristike memorijskih sistema

- 3. Jedinica prenosa

- Za unutrašnju memoriju može biti jednaka dužini reči, ali je obično duža npr. 64, 128 ili 256 bita
- Jednaka je broju električnih veza koje ulaze u memorijski modul ili iz njega izlaze
- **Reč** – je „prirodna“ jedinica organizacije memorije i obično je jednaka broju bitova potrebnih za predstavljanje celog broja i dužini instrukcije
 - Intelx86 procesori imaju raznovrsne dužine instrukcija (iskazane u umnošcima bajtova) i ima reči veličine 32 bita
- **Adresibilna jedinica** – u nekim sistemima to je reč
 - Neki sistemi dozvoljavaju adresiranje na nivou bajtova
- **Jedinica prenosa**
 - Za unutr. mem. = je broju bitova koji se mogu istovremeno pročitati iz memorije ili upisati u memoriju
 - Za spolj. mem. podaci se obično prenose u većim jedinicama tzv. blokovi podataka

Ključne karakteristike memorijskih sistema

- 4. Metode pristupanja jedinicama podataka
 - 1. Sekvencijalno pristupanje
 - 2. Direktno
 - 3. Nasumično
 - 4. Asocijativno

Metode pristupanja memoriji

- 1. Sekvencijalno pristupanje

- Pristupanje mora da se obavlja određenim linearnim redom
- Memorija je organizovana u jedinice podataka tzv. zapise
- Informacije o adresama se koriste za razdvajanje zapisa
- Zajednički mehanizam za čitanje i upisivanje, pri čemu se vrši pomeranje sa trenutnog na željeni položaj preskakanjem svih zapisa između
- Vreme pristupa je promenljivo
- Magnetne trake koriste sekvencijalno pristupanje

Metode pristupanja memoriji

- 2. Direktno pristupanje
 - Zajednički mehanizam za čitanje i upisivanje
 - Pojedinačni blokovi ili zapisi imaju jedinstvenu adresu na osnovu fizičke lokacije
 - Pristupanje se obavlja direktnim pristupom blizu traženih podataka, a zatim se pretražuje kako bi se stiglo do konačne lokacije
 - Vreme pristupa je promenljivo
 - Koriste ga magnetni diskovi

Metode pristupanja memoriji

- 3. Nasumično pristupanje

- Svaka lokacija u memoriji koja može da se adresira ima zaseban fizički ožičeni mehanizam za adresiranje
- Vreme pristupa za određenu lokaciju ne zavisi od prethodnih pristupanja i nepromenljivo je
- Bilo koja lokacija može da se direktno adresira i da joj se direktno pristupi
- Koriste ga glavna memorija i neki sistemi keš memorija

Metode pristupanja memoriji

- 4. Asocijativno pristupanje
 - Tip memorije sa nasumičnim pristupom, poređenjem željenih lokacija bita unutar reči pronalazi se podudarnost (ovo se radi za sve reči istovremeno)
 - Reč se preuzima na osnovu dela njenog sadržaja, a ne na osnovu njene adrese
 - Vreme pristupa određenoj lokaciji ne zavisi od lokacije i prethodnih pristupa tj. nepromenljivo je
 - Koriste ga **keš memorije**

Ključne karakteristike memorijskih sistema

- 5. Performanse memorije

- Vreme pristupa (kašnjenje)

- Za memoriju sa nasumičnim pristupom to je vreme potrebno da se obavi operacija čitanja ili upisivanja

- Trajanje memorijskog ciklusa

- Odnosi se na memoriju sa nasumičnim pristupom
- Vreme pristupa + dodatno vreme kako bi sledeći pristup mogao da počne (prelazna stanja na signalnim linijama)

- Brzina prenosa

- Brzina kojom se podaci prenosu u/iz memorije
- Sa nasumičnim pristupom jednako je $\frac{1}{\text{trajanje ciklusa}}$

Ključne karakteristike memorijskih sistema

- 6. Fizička realizacija memorije
 - Poluprovodnička
 - Magnetna
 - Optička
 - Magnetno-optička

Ključne karakteristike memorijskih sistema

- 7. Fizičke karakteristike memorije

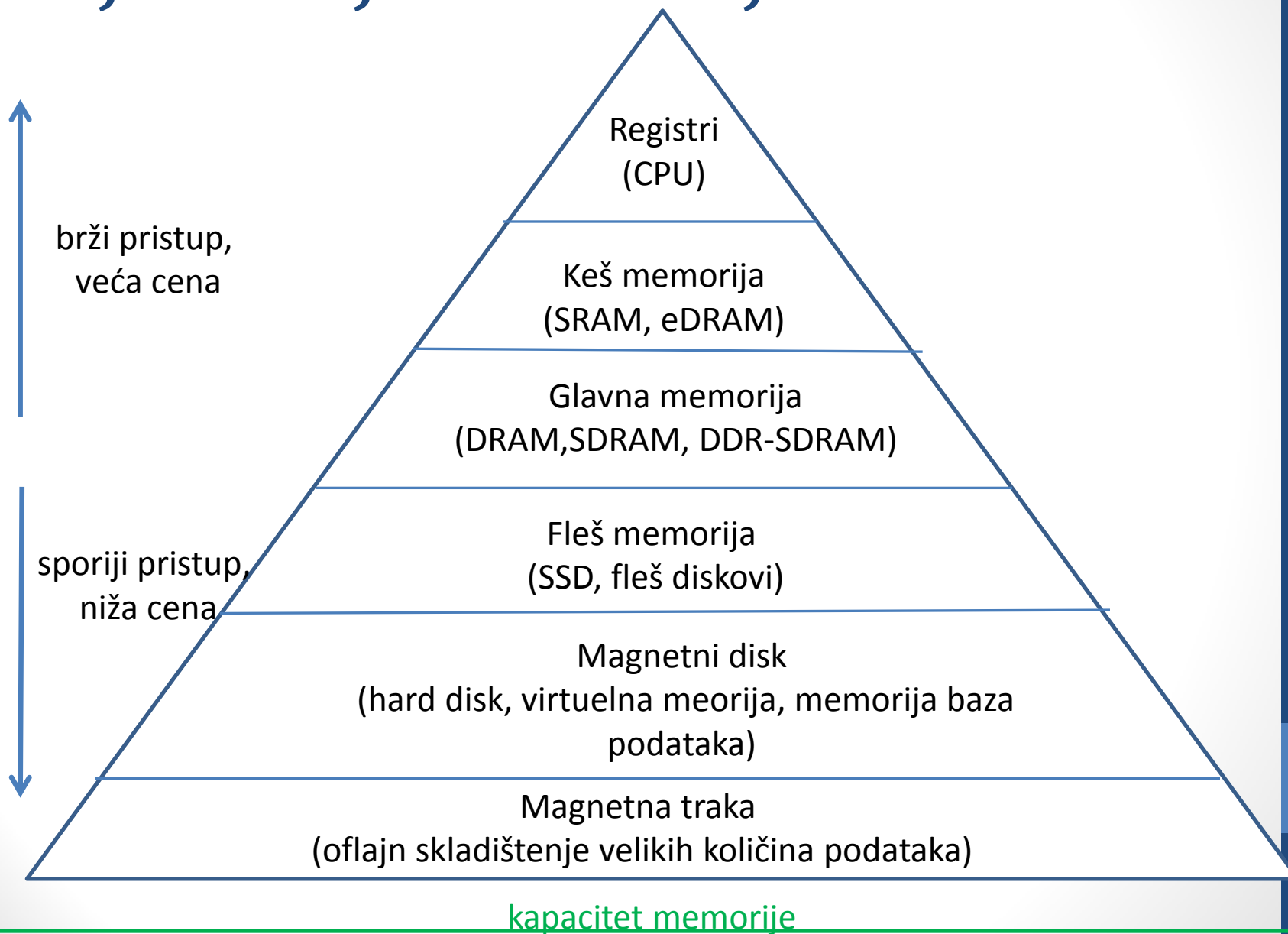
- **Postojana/ nepostojana**

- Nepostojana: podaci se gube kad se isključi električno napajanje
- Postojana: memorije sa namagnetisanom površinom (informacije se ne gube nakon isključenja napajanja)

- **Obrisiva / neobrisiva**

- Neobrisiva: memorija koja ne može ni da se briše ni da se menja (npr. ROM (read-only memory))

Hijerarhija memorije



Hijerarhija memorije

- Kretanjem naniže kroz hijerarhiju memorije dešava se sledeće:
 - Smanjuje se cena po bitu
 - Povećava se kapacitet
 - Povećava se vreme pristupa
 - Procesor ređe pristupa memoriji
 - Ovo se postiže iskorišćenjem principa lokalnosti

Hijerarhija memorije

- Manje, skuplje i brže memorije se dopunjavaju većim, jeftinijim i sporijim memorijama
- Moguće je organizovati podatke duž ove hijerarhije memorije tako da procenat pristupa za svaki naredni nivo bude značajno manji od nivoa iznad
- Tipičan računarski sistem je opremljen hijerarhijom memorijskih podсистема, od kojih su neki unutrašnji (procesor im direktno pristupa), a neki su spoljašnji (procesor im pristupa preko U/I modula)

Princip lokalnosti

- **Princip lokalnosti** ili **lokalnost pozivanja memorije** – memorija koju procesor poziva tokom izvršavanja programa, kako za instrukcije tako i za podatke, teži tome da se grupiše
 - primeri: programske petlje, potprogrami koji se ponavljaju; operacije sa nizovima, matricama

Princip lokalnosti

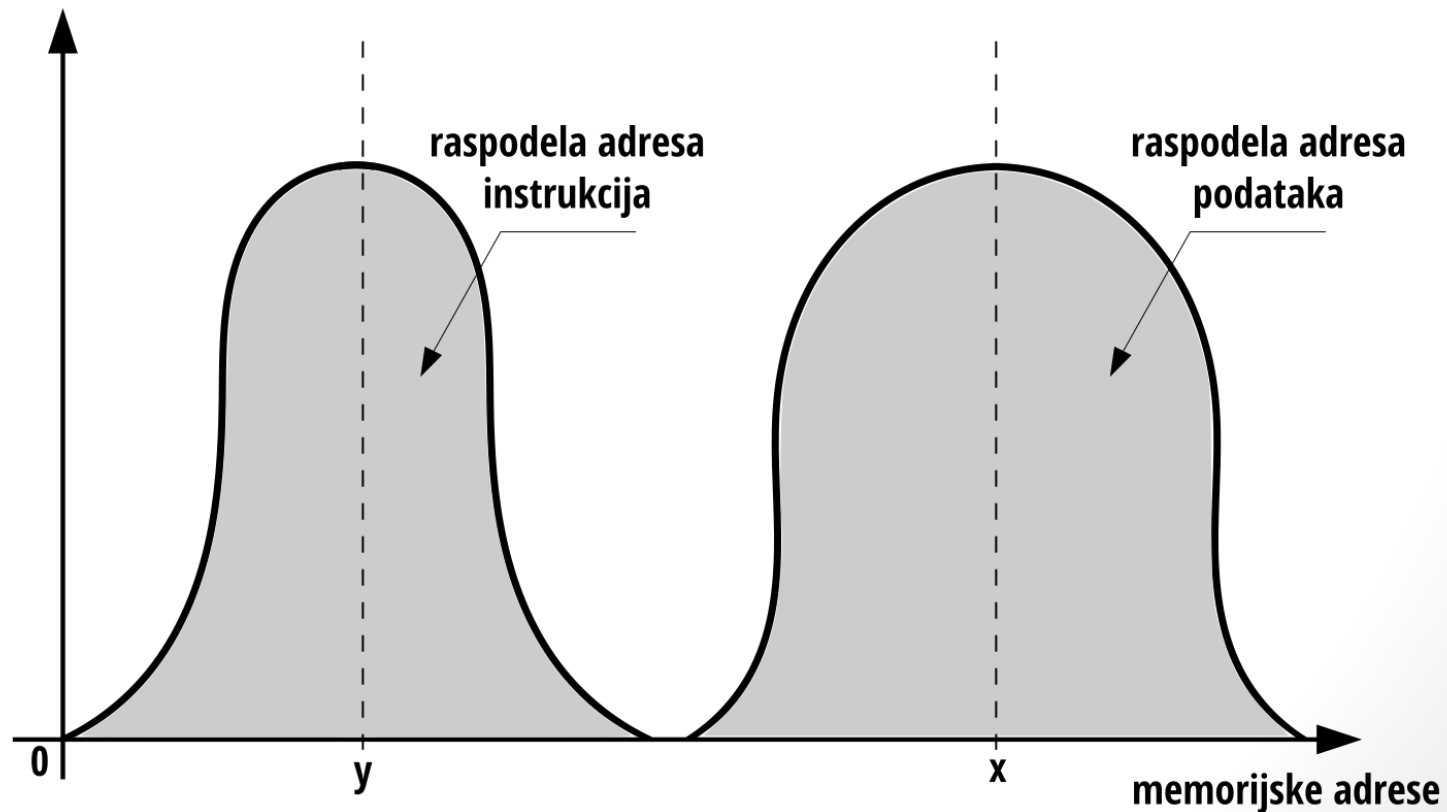
- Dva oblika lokalnosti:
- **1. Vremenska lokalnost** – sklonost programa da u bliskoj budućnosti poziva one jedinice memorije koje je pozivao u neposrednoj prošlosti
- **2. Prostorna lokalnost** – sklonost programa da pristupa jedinicama memorije čije su adrese jedna blizu druge tj. odražava sklonost procesora da instrukcijama pristupa redom kao i da lokacijama podataka pristupa redom (npr. obrada tabele podataka)
- Oba oblika lokalnosti odnose se i na instrukcije i podatke u memoriji:
 - Vremenska lokalnost podataka i vremenska lokalnost instrukcija
 - Prostorna lokalnost podataka i prostorna lokalnost instrukcija

Princip lokalnosti

- Primer:
- Vremenska lokalnost se iskorišćava tako što se nedavno korišćene instrukcije i podaci čuvaju u keš memoriji
- Prostorna lokalnost se iskorišćava:
 - upotrebom većih blokova keš memorije, i
 - uključivanjem mehanizma za preuzimanje unapred u keš memoriju

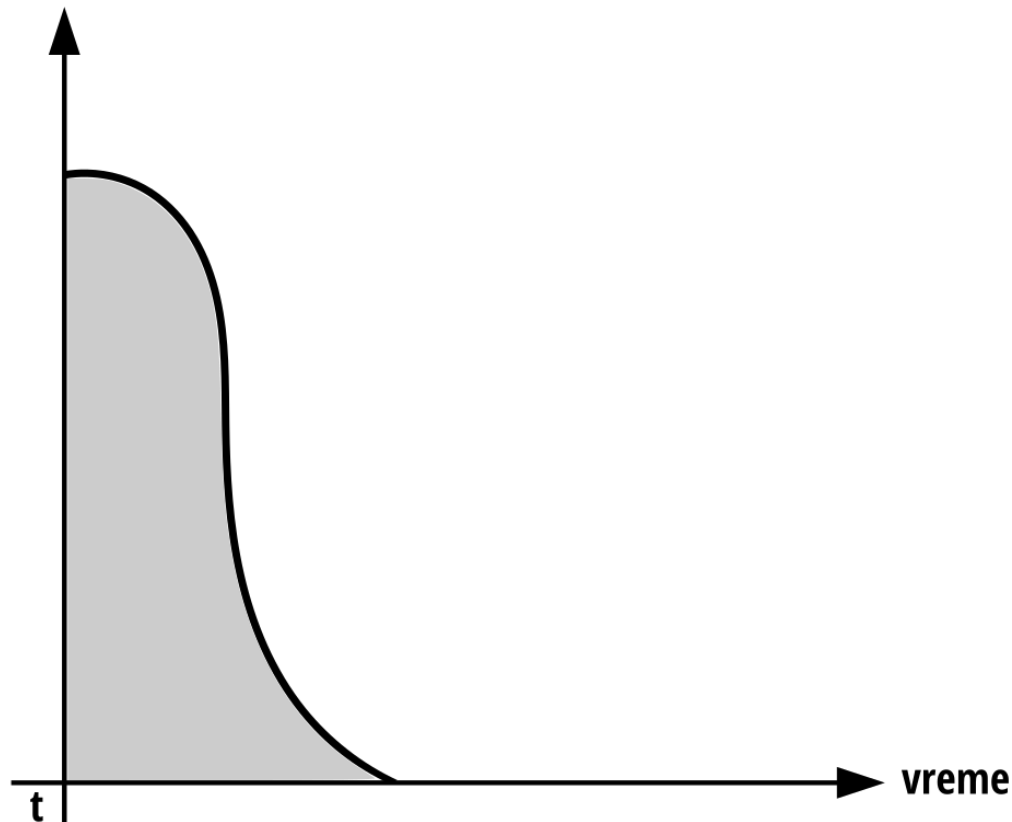
Idealizovano ponašanje prostorne lokalnosti

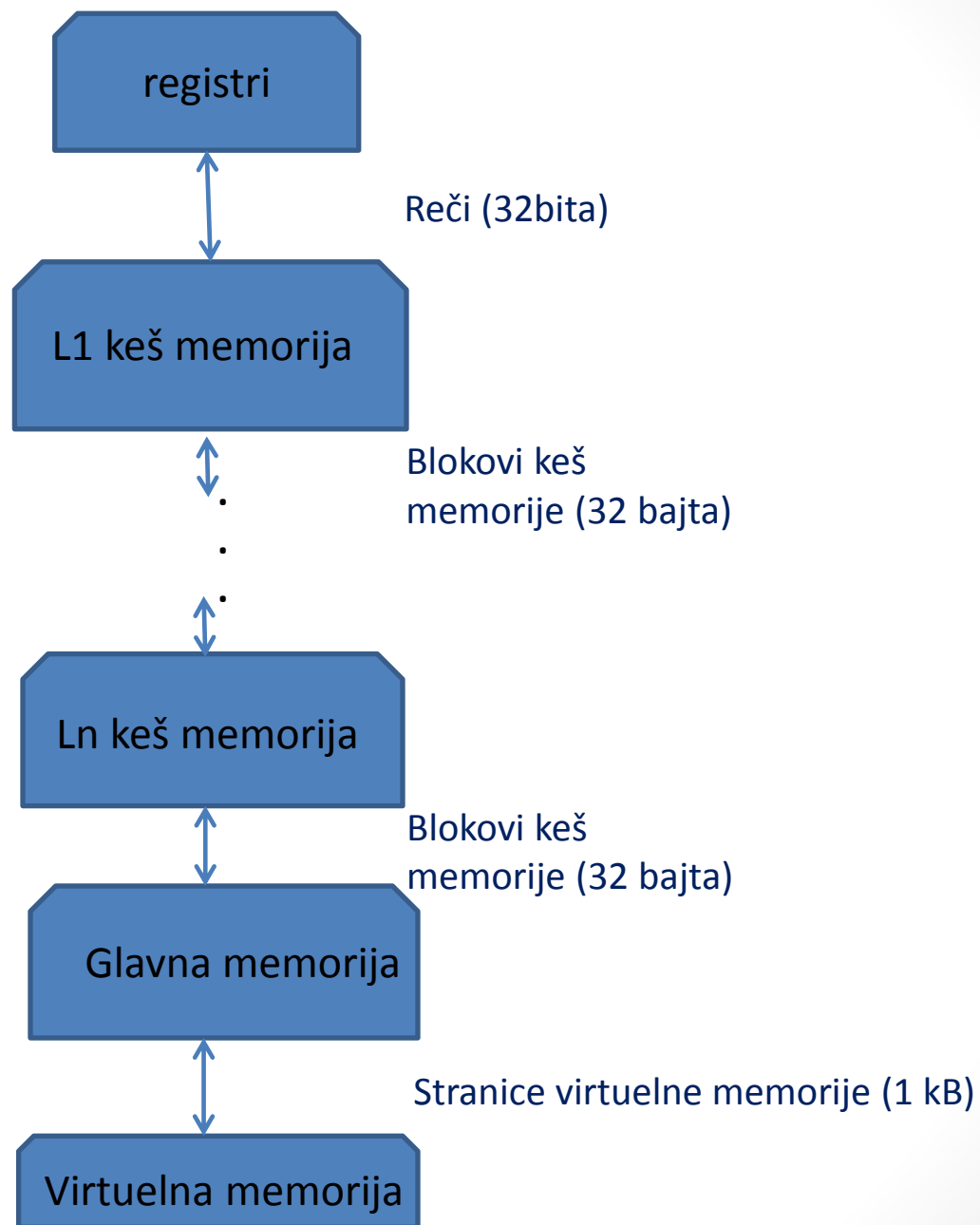
- Prikaz raspodela verovatnoće za sledeće pristupanje memoriji, pri čemu je poslednje pristupanje podacima u memoriji na lokaciji x , a poslednje preuzimanje instrukcije sa lokacije y



Idealizovano ponašanje vremenske lokalnosti

- Prikaz raspodele verovatnoće pristupa memoriji u odnosu na jedinicu memorije kojoj je pristupljeno u trenutku t





Članovi hijerarhije memorije

- **Registri** u procesoru: najbrži, najmanji i najskuplji tip memorije
 - Tipično, procesor obuhvata nekoliko desetina registara, ali može i stotinak registara
- Između CPU registara i glavne memorije stoji keš memorija
- **keš memorija** je organizovana u više nivoa
 - Keš memorija nivoa 1, L1 keš memorija (Level 1 cache) – najbliža je registrima procesora
 - Skoro uvek podeljena na keš memoriju za instrukcije i keš memoriju za podatke
 - Keš memorija nivoa 2, L2 keš memorija
 - Sledeći nivo keš memorije
 - I za nju važi podela na keš memoriju za instrukcije i keš memoriju za podatke

Članovi hijerarhije memorije

- L3 keš memorija i L4 keš memorija
 - Obično se ne dele na zasebne delove za instrukcije i podatke
 - Mogu da budu zajedničke za više procesora
- Keš memorije su tipično izrađene u SRAM (*static random access memory*) tehnologiji
- Viši nivoi keš memorije se ostvaruju korišćenjem ugrađene dinamičke memorije sa nasumičnim pristupom eDRAM (*embedded dynamic RAM*)

Članovi hijerarhije memorije

- Glavna memorija
 - Najvažnija unutrašnja memorija računara
 - Svaka lokacija ima jedinstvenu adresu
 - Vidljiva je programeru, dok keš memorija nije
 - Razne nivoe keš memorije kontroliše hardver i koristi se kao podrška premeštanju podataka između glavne memorije i registara procesora kako bi se poboljšale performanse

Članovi hijerarhije memorije

- Registri, keš memorija i glavna memorija
 - Obično su to nepostojane memorije
 - Koriste poluprovodničku tehnologiju izrađenu u više tipova koje se razlikuju po brzini i ceni
- Podaci se trajnije čuvaju na spoljašnjim uređajima za skladištenje velike količine podataka
 - Hard diskovi, magnetni diskovi i trake, optički diskovi
- Spoljašnja, postojana memorija se naziva **sekundarna (pomoćna) memorija**

Članovi hijerarhije memorije

- Sekundarna memorija
 - Koristi se za čuvanje datoteka sa programima i podacima
 - Programerima je vidljiva u obliku datoteka i zapisa, ne u vidu pojedinačnih bajtova i reči
 - Hard diskovi se koriste kao proširenje glavne memorije poznato i kao **virtuelna memorija**
 - Pristupa joj se putem U/I modula

Principi projektovanja hijerarhije memorije

- 3 principa utiču na projektovanje hijerarhije memorije:
 - 1. **Lokalnost**
 - 2. **Inkluzija** – sve informacije se prvo skladište na nivou M_n , gde je n najudaljeniji nivo od procesora
 - Tokom obrade podskup nivoa M_n se kopira na nivo M_{n-1} , zatim se deo nivoa M_{n-1} kopira na nivo M_{n-2} itd.
 - 3. **Koherencija** – kopije istih podataka u susednim nivoima memorije moraju da budu potpuno identične

Koherencija memorije

- Ako se u keš memoriji promeni neka reč, kopije te reči na svim višim nivoima moraju da se ažuriraju odmah ili čim je moguće
- Primer: kod računara IBM z13 osam L2 keš memorija dele istu L3 keš memoriju, a tri L3 keš memorije dele istu L4 keš memoriju
 - **Vertikalna koherencija**- ako neko jezgro procesora promeni blok podataka u L2 keš memoriji, to ažuriranje treba da se vrati na nivo L3 pre nego što druga L2 keš memorija preuzme taj blok podataka
 - **Horizontalna koherencija** – ako dve L2 keš memorije (koje dele istu L3 keš memoriju) imaju kopije istog bloka podataka, tada ako se taj blok promeni u jednoj od L2 keš memorije, druga L2 keš memorija mora da se upozori da joj je kopija zastarela

Pristupanje memoriji sa dva nivoa

- Uzećemo kao najjednostavniji primer, koji se retko primenjuje u savremenim sistemima, računar koji ima samo jedan nivo keš memorije povezan sa glavnom memorijom
 - **Keš memorija** – kao posredna (eng.buffer) memorija između procesora i glavne memorije -> nastaju dva nivoa unutrašnje memorije
 - **Mehanizam keširanja glavne memorije** ostvaruje se u hardveru i obično je nevidljiv za operativni sistem

Pristupanje memoriji sa dva nivoa

- Nivo memorije M_1 je manji, brži i skuplji u odnosu na nivo memorije M_2
- Nivo M_1 se koristi kao privremeno skladište za deo sadržaja većeg nivoa M_2
- Kada se pozove memorija, prvo se pokuša pristupanje određenom podatku na nivou M_1 . Ako uspe, to je brz pristup
- Ako ne uspe, tada se blok memorijskih lokacija kopira sa nivoa M_2 na nivo M_1 , a zatim se traženom podatku pristupa na nivou M_1
- Zbog principa lokalnosti, pošto se blok podataka preneo na nivo M_1 , u narednom periodu očekuje se više pristupa lokacijama u tom bloku -> brži pristup

Pristupanje memoriji sa dva nivoa

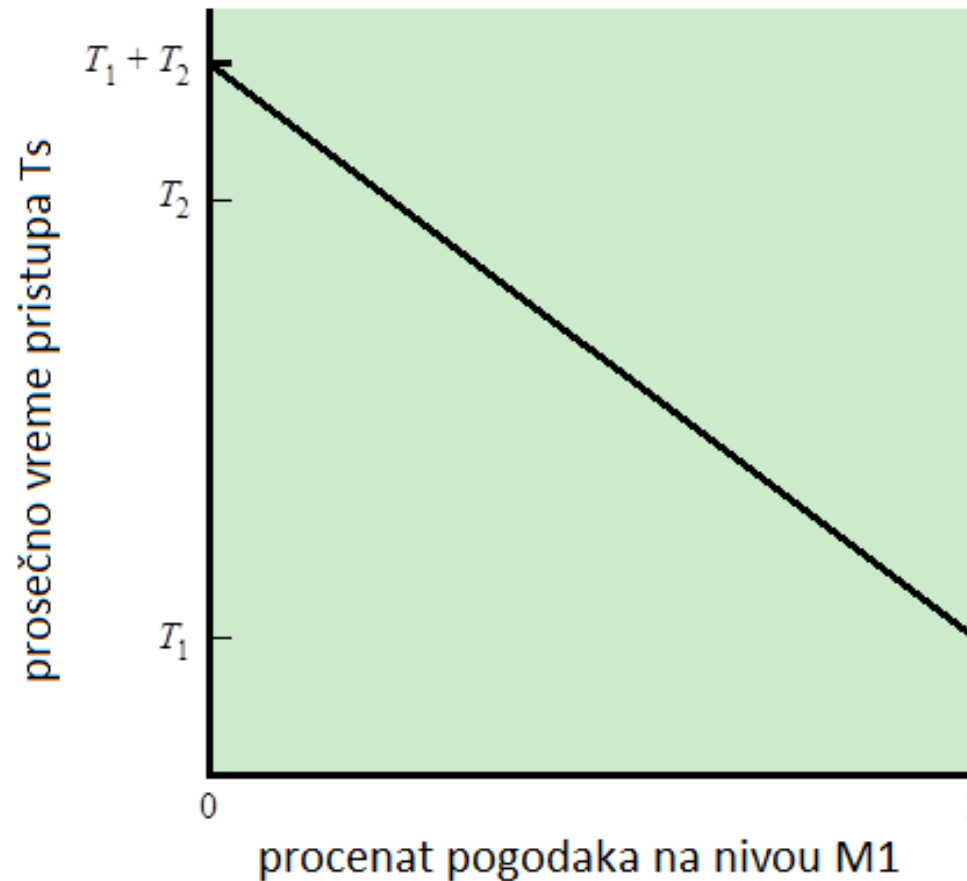
- Prosečno vreme pristupa nekom podatku:

$$T_s = H * T_1 + (1 - H) * (T_1 + T_2)$$

- T_s - prosečno vreme pristupa za sistem
- T_1 - vreme pristupa za nivo M_1 (npr. keš memorija)
- T_2 - vreme pristupa za nivo M_2 (npr. glavna memorija)
- H – stopa pogodaka (eng. *hit*). Pogodak je slučaj kada se podatak nađe u bržoj memoriji M_1

Pristupanje memoriji sa dva nivoa

- Performanse pristupa na nivou M_1



Pristupanje memoriji sa dva nivoa

- Ako postoji jaka lokalnost moguće je ostvariti velike stope pogodaka i sa relativno malom veličinom keš memorije
- Keš memorija od 1K do 128K reči je uopšteno odgovarajuća, dok se glavna memorija izražava u gigabajtima
- Zaključak: relativno mali nivo M_1 daje velike stope pogodaka zahvaljujući lokalnosti

Pristupanje memoriji sa više nivoa

- Primer memorije sa 3 nivoa: jedan nivo keš memorije M_1 , glavna memorija M_2 i sekundarna memorija M_3
- Prosečno vreme pristupa nekom podatku:

$$Ts = h_1 * t_1 + (1 - h_1) * h_2 * t_2 + (1 - h_1) * (1 - h_2) * h_3 * t_3$$

- h_i - stopa pogodaka na nivou M_i (uslovna verovatnoća da se podaci nalaze na nivou M_i pod uslovom da se ne nalaze na nivou M_{i-1})
- t_i - ukupno vreme potrebno za pristupanje podacima na nivou M_i . Jednako je zbiru svih vremena na putanji do pogotka na nivou M_i

Pristupanje memoriji sa više nivoa

