

# Gospodarenje memorijskim prostorom

---

**Barbieri, Alberto**

**Undergraduate thesis / Završni rad**

**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Pula / Sveučilište Jurja Dobrile u Puli**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:137:038471>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-20**



*Repository / Repozitorij:*

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Odjel za informacijsko komunikacijske tehnologije

**ALBERTO BARBIERI**

**GOSPODARENJE MEMORIJSKIM PROSTOROM**

Završni rad

Pula, 2016.

Sveučilište Jurja Dobrile u Puli  
Odjel za informacijsko komunikacijske tehnologije

**ALBERTO BARBIERI**

**GOSPODARENJE MEMORIJSKIM PROSTOROM**

Završni rad

**JMBAG: 0318000733, redovni student**

**Studijski smjer: Preddiplomski sveučilišni studij Informatika**

**Predmet: Osnove IKT**

**Mentor: Prof. dr. sc. Vanja Bevanda**

Pula, rujan 2016.



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Alberto Barbieri, kandidat za prvostupnika Preddiplomskog sveučilišnog studija Informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Alberto Barbieri

---

U Puli, rujan 2016.



## IZJAVA

### o korištenju autorskog djela

Ja, Alberto Barbieri dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Gospodarenje memorijskim prostorom“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan 2016.

Alberto Barbieri

---

## SADRŽAJ

|   |    |
|---|----|
| UVOD .....  | 1  |
| 1. MEMORIJA .....   | 3  |
| 1.1. KAKO DIJELIMO MEMORIJU .....   | 6  |
| 1.2. LOGIČKA ORGANIZACIJA MEMORIJE .....                                  | 10 |
| 2. OPERACIJSKI SUSTAV I NJEGOVE FUNKCIJE .....                            | 13 |
| 3. OSNOVE GOSPODARENJA MEMORIJSKIM PROSTOROM.....                         | 16 |
| 3.1. MEMORIJSKO-UPRAVLJAČKA JEDINICA (MMU) .....                          | 17 |
| 3.1.1. FIZIČKI I LOGIČKI ADRESNI PROSTOR I VEZIVANJE ADRESA.....          | 17 |
| 3.1.2. ORGANIZACIJA LOGIČKOG ADRESNOG PROSTORA.....                       | 18 |
| 3.1.3. ALOKACIJA MEMORIJE – DODJELA MEMORIJE PROCESIMA .....              | 19 |
| 3.2. GLAVNE FUNKCIJE I ZAHTJEVI SUSTAVA ZA GOSPODARENJE<br>MEMORIJOM..... | 20 |
| 4. PROBLEMI UPRAVLJANJA MEMORIJOM .....                                   | 22 |
| 4.1. UNUTARNJA I VANJSKA FRAGMENTACIJA .....                              | 22 |
| 4.2. SLOBODNE MEMORIJSKE PARTICIJE.....                                   | 26 |
| 4.3. PODRŠKA ZA DINAMIČKO PUNJENJE I VEZIVANJE.....                       | 28 |
| 5. PROŠIRENJE MEMORIJSKOG PROSTORA.....                                   | 30 |
| 6. ZAKLJUČAK .....  | 31 |
| LITERATURA.....   | 33 |
| POPIS SLIKA .....   | 34 |
| SAŽETAK.....  | 35 |
| SUMMARY .....   | 37 |

## UVOD

U radu dan je prikaz osnova gospodarenja memorije, njezin značaj i važnost uloge koju ima u cijelom računalnom sustavu današnjeg informatičkog doba. Napretkom informatičke tehnologije proces upravljanja memorijom se drastično poboljšao kako hardverski dio tako i softverski. Nastali nedostaci koji su postojali te problemi koji su se tako javljali, jasno su pokazali da se velika pažnja mora usmjeriti na njeno gospodarenje. Do sada bila su nastojanja kako maksimalno iskoristiti memoriju, ali sa što manje troškova i čim većom brzinom korištenja. Razvoj informacijsko – komunikacijskih tehnologija dostigao je vrhunac razvoja, te vidimo da u praksi informatički stručnjaci željeli bi imati na raspolaganju beskonačno veliku i brzu memoriju na svojim računalnim sustavima.

Nakon uvoda, u prvom poglavlju dan je prikaz osnovnih pojmova memorije, kako dijelimo memoriju i logičku organizaciju memorije. Memorija je neophodna za normalno funkcioniranje suvremenih računalnih sustava i čini njihov centralni najvažniji dio sustava.

U drugom poglavlju ukratko je opisano što je to operacijski sustav te koje zadatke on mora izvršiti, a to su olakšanje uporabe računala i organizacija djelotvornog iskorištavanja svih dijelova računala. Suvremeni računalni sustavi upotrebljavaju procesore s takvim spremničkim međusklopovima, koji omogućuju da operacijski sustavi djelotvorno gospodare spremničkim memorijskim prostorom.

U sljedećem trećem poglavlju navedene su osnove gospodarenja memorijskim prostorom te je dan prikaz što je to memorijsko-upravljačka jedinica, zatim što je fizički i logički adresni prostor te vezivanje adresa. Ukratko opisana je alokacija memorije tj. kako se dodjeljuje memorija procesima. Navedene su glavne funkcije i koji su osnovni zahtjevi sustava za gospodarenje memorijom.

U četvrtom poglavlju navedeni su osnovni problemi upravljanja memorijom, što je to unutarnja i vanjska fragmentacija, slobodne memorijske particije i podrška za dinamičko punjenje i vezivanje. Korisnički programi i sve veći broj raznovrsnih aplikacija u današnje doba postaju sve više složeniji i zahtjevniji, budući da sve više nedostaje količine memorije koja je dodijeljena programu. Osnovno pitanje cijelog informacijskog napretka postaje kako danas povećati količinu memorije u računalima, budući da količina podataka koja se može pohraniti u memoriju postaje sve veća. S aspekta gospodarenja memorijskim prostorom, informatički stručnjaci susreću se sa problemima dijeljenja memorije i usklađivanja brzina rada procesora i glavne memorije.

U narednom petom poglavlju ukratko je opisano što je to proširenje memorijskog prostora te dan je naglasak na relokaciji odnosno defragmentaciji i swap radne memorije. Računalni sustavi današnjice vane za neprestanim proširenjem memorijskog kapaciteta i što većeg ubrzanja pohrane podataka. Za proširenje memorijskog prostora koristi se relokacija, koja znači premjestiti odnosno odrediti novi položaj određenog programa u memoriji.



## 1. MEMORIJA

Danas živimo u vremenu brzog razvoja informacijsko – komunikacijskih tehnologija u svim područjima ljudskog života. Kako informacijska tehnologija neprestano napreduje, tako paralelno napreduju i sve veći korisnički zahtjevi, a naročito zahtjevi u pogledu gospodarenja memorijskim prostorom.

**Memorija** (engl. memory) čini centralni dio suvremenih računala, koji je neophodan za njegovo normalno funkcioniranje. Možemo je definirati kao sredstvo za spremanje podataka i jedna je od temeljnih dijelova računalnog sustava. Kao funkcijska jedinica digitalnog računalnog sustava ona prima, čuva i predaje digitalne podatke (engl. digital data). U memoriji se mogu nalaziti spremljene naredbe, programi, podaci koji su već obrađeni ili podaci koji tek trebaju biti obrađeni, zatim pomoćni rezultati obrade i međurezultati te konačni rezultati prije nego što se predaju izlaznoj jedinici (output). Svaka lokacija u memoriji jednoznačno je određena adresom (engl. address) kojom se može pristupiti (engl. access) toj memorijskoj lokaciji i s nje točno pročitati određeni pohranjeni podatak ili pohranjeni novi podatak. Znači, memorija kao jedna od osnovnih dijelova računala, jedno je veliko polje okteta (skup od osam bitova, engl. byte) ili riječi, koji imaju svaki svoju adresu, gdje procesor dohvaća naredbe iz memorije, a rezultati tih naredbi mogu biti dodatni pristupi memoriji kako bi se dohvatilo operande (dio naredbe koji određuje radnju što će se obaviti nad određenim podatkom – operand) i pohranilo rezultate operacija.

U osnovne pojmove memorije spada **adresa memorije** (engl. memory address) ili memorijska adresa, koja je pozitivan cijeli broj koji prikazuje mjesto na kojem se u izravnoj memoriji (engl. random access memory) nalazi određeni podatak, odnosno to je broj koji jednoznačno omogućava dohvat određenog podatka.

**Memorijski ciklus čitanja** podrazumijeva slijedeće radnje koje izvodi procesor kada želi dohvatiti podatak, tj. opisan je pojednostavljen tijekom događaja. Procesor na adresnu sabirnicu stavlja adresu podatka. To je uređaj ili elektronički sklop, koji može izvoditi određene akcije nad podacima. On šalje memoriji određeni signal da je na adresnoj sabirnici ispravna adresa (engl. address strobe signal), odnosno to podrazumijeva memoriju koja se može adresirati. Memorija na podatkovnu sabirnicu postavlja podatak, koja šalje procesoru signal da je traženi podatak postavljen na podatkovnu sabirnicu (engl. data strobe signal). Na kraju procesor dohvaća podatak sa podatkovne sabirnice u registar.

**Brzina memorije** ili vrijeme pristupa podrazumijeva određeno vrijeme koje je potrebno da procesor iz memorije dohvati podatak. Budući da su pozivi procedura vremenski vrlo zahtjevne operacije, najznačajnija pozornost pridaje se mehanizmima za ubrzanje pozivanja. Kada procesor zatraži određeni podatak iz memorije, ne radi ništa osim što čeka da memorija isporuči podatak.

**Registri** (engl. registry) su „memorija“ koja je prisutna na samom čipu procesora. Ona podrazumijeva središnju hijerarhijsku bazu podataka u novijim Windows operacijskim sustavima (engl. hierarchical database), a služi za pohranu podataka o ustrojavanju sustava za jednog korisnika ili više (engl. user, multiuser system). Ovdje se nalaze profili svih korisnika sustava, podaci o programskim instalacijama u sustavu, resursi mrežne komunikacije itd. Važna strana registara je da oni rade na brzini procesora, tj. nema određenog vremena pristupa te procesor odmah započinje obradu određenog podatka, jer nema čekanja da podatak stigne. U registrima trebalo bi držati sve podatke, a budući da se svi podaci ne drže u registrima te se neki moraju smjestiti u memoriju, to vidno usporava rad programa. Tu se pokazuje njihova loša strana, jer ih ima jako malo, ovisno o određenom procesoru (najčešće manje od 128).

**MMU** (engl. memory management unit) je jedinica za upravljanje memorijom, a nazivamo je memorijsko – upravljačka jedinica. Ona podrazumijeva dio računalnog sklopovlja odgovornog za upravljanje pristupom računalnom memorijom koju je zahtijevala središnja jedinica. To je hardverska komponenta, odnosno dio procesora koji prevodi logičke (virtualne) adrese u fizičke adrese. U njezine funkcije spada provoditi prividne adrese u stvarne adrese (tj. upravljanje prividnom memorijom), zaštita memorije, kontroliranje priručne memorije, arbitriranje sabirnice te kod jednostavnih računalnih arhitektura (naročito kod 8-bitnih sustava) prespajanje blokova (engl. bank switching).

**Page tables** jesu strukture podataka. Podaci se obično pohranjuju u tablici koja se naziva **tablica okvira** ili **frame table**, koja posjeduje određeni redak za svaki okvir koji pokazuje da li je okvir slobodan ili ne i ako nije koji ga proces ili procesi koriste.

Operacijski sustav za njih rezervira dio RAM-a (engl. random access memory, to je memorija s izravnim pristupom, memorija s direktnim pristupom). Ova memorija u kojoj se sve lokacije mogu dosegnuti u danom vremenu, označava upisno-ispisne memorije koja omogućuje čitanje i upisivanje novoga sadržaja pomoću kojih MMU radi translaciju virtualnih adresa u fizičke adrese. Translacija se radi za intervale adresa veličine jednog page frame-a, te je u page tablici

potreban samo jedan podatak od 32 bita za cijeli niz od 4KB adresa ili koliko je već veličina page frame-a.

**Swap** podrazumijeva unaprijed rezervirani prostor na disku na koji operacijski sustav sprema dijelove „neaktivnih“ programa kada se dogodi da ponestane fizičke memorije. To prilično usporava rad ovih programa kada opet pristupe podacima na disku, budući da je disk nekoliko redova veličine sporiji od RAM-a.

**Buffer cache** podrazumijeva dio RAM-a koji operacijski sustav rezervira radi bržeg čitanja sa diska. Operacijski sustav najprije provjerava da li se traženi sektor već nalazi u buffer cache-u te ako ga tamo ne nalazi, tek onda dohvaća se sa diska, koji je mnogo sporiji od RAM-a. Veličina buffer cache-a nije fiksna kategorija, jer ju operacijski sustav dinamički prilagođava prema ukupnoj trenutnoj potražnji memorije.

**BIT** je najmanja fizička jedinica upisa podatka, zauzima jedno mjesto u memoriji računala, ima vrijednost 0 ili 1.

**BAJT** (engl. byte) je fizička jedinica podataka koja se sastoji od 8 bitova, predstavlja logički 1 znak, npr. slovo ili broj ili drugo.

**Riječ** je osnovna fizička jedinica pohrane podatka na računalu, može biti 8-bitna, 16-bitna ili 32-bitna.

U današnje vrijeme gdje su informacijsko-komunikacijske tehnologije dostigle vrhunac razvoja, moguće je imati velik broj pokrenutih programa, koji zauzimaju puno više memorijskog prostora nego što je to fizički raspoloživo. U mnoštvu današnjih sve složenijih aplikacija, gdje se zahtijeva sve više memorije taj princip vrijedi, premda se cijeli sistem rada jako ne usporava. Velika je vjerojatnost da će se potreban podatak zatražiti u narednom vremenskom periodu sa diska, RAM-a ili swapa. Informatičari željeli bi imati na raspolaganju beskonačno veliku i beskonačno brzu memoriju na svojim računalima, koja ne gubi svoj sadržaj kada se desi da joj se uskrati napajanje. Zato u današnje vrijeme imamo podijeljenu memoriju prema hijerarhiji, a svaki operacijski sustav ima zadatak koordinirati upotrebu tih memorija.

### 1.1. KAKO DIJELIMO MEMORIJU

Centralna procesna jedinica je u računalnoj prošlosti bila uvijek brža nego glavna radna ili primarna memorija. U današnje vrijeme računala imaju odnos između memorijskog ciklusa prema procesorskom 3 do 10:1, što bi značilo da postoji velika neprikladnost u odnosima između brzine procesora i brzine memorije. Ovaj glavni problem zove se **latentnost memorije** (engl. memory latency), a prema Ribarić, S., str. 260. „latentnost se izražava vremenom koje protekne između zahtjeva što ga procesor upućuje memoriji i primitka podatka iz memorije“. „Arhitekti i tehnolozi znaju i mogu graditi memorijske module koji imaju jednaku brzinu kao i centralna procesna jedinica, no cijena takvim memorija (kapaciteta više desetaka ili stotina MB) znatno bi povećala ukupnu cijenu sustava i pogoršala odnos performanse i cijene.“ Tako se može zaključiti da je tehnološki činitelj posljedica ekonomskog činitelja, a posljedice tehnoloških činitelja utječu na latentnost memorije suvremenih računala. Drugi prisutan problem koji se pojavljuje u memorijskom sustavu suvremenih računala je **kapacitet glavne memorije**, gdje „veličina fizički izvedene memorije (reda MB) također izravno utječe na odnos performansa/cijena, to više što brojne primjene zahtijevaju vrlo velike kapacitete memorije (reda GB i više)“ (prema Ribarić, S., str. 260.).

Ovaj prisutan problem latentnosti memorije i kapaciteta glavne memorije informatički stručnjaci rješavaju upotrebom **memorijske hijerarhije**, te se primjenom ovog arhitektonskog koncepta postiže slijedeći cilj: „Glavna ili primarna memorija prividno se pojavljuje kao memorija koja ima kapacitet sekundarne memorije (dakle vrlo veliki kapacitet, reda GB), a brzinu jednaku brzini najbrže ili gotovo najbrže memorije u memorijskoj hijerarhiji, odnosno brzini centralne procesne jedinice. Ostvarenje ovog cilja omogućeno je primjenom načela **virtualne memorije i priručne ili cache memorije**“ (prema Ribarić, S., str. 261.).

Današnja računala prema kapacitetu memorije i njezine brzine, možemo na temelju memorijske strukture koje susrećemo u računalnom sustavu razvrstati tako da čine **hijerarhijsko uređenje**.

**Lokalna memorija** ili **registri** imaju brzinu istog reda veličine kao i sklopovi procesora i čine logičku i najčešće i fizičku cjelinu sa samim procesorom. Kapacitet memorije je od 8 do 64 registra. Iznimka su posebno procesori arhitekture RISC koji imaju stotinu ili čak više stotina registara.

**Priručna ili cache memorija** je većeg kapaciteta od lokalne memorije te nije sporija. Ova memorija je za korisnika nevidljiva tj. skrivena (riječ cache je francuskog porijekla /caher/, označava skrivanje ili skrovito mjesto za pohranjivanje). Ova memorija može biti izvedena kao

fizička cjelina sa samim procesorom. Priručna memorija može funkcionalno biti izvedena kao memorija za pohranjivanje instrukcija i podataka (tzv. I&D cache), kao izdvojena instrukcijska priručna memorija (I cache), kao izdvojena priručna memorija podataka (D cache). Kapacitet memorije kreće se od 4 KB do 64 KB, a nekad i više.

**Glavna, radna ili primarna memorija** kapaciteta je reda MB i sporija je u odnosu na priručnu memoriju.

**Sekundarna memorija ili disk memorija** je vrlo velikog kapaciteta (reda GB) i relativno je spora u odnosu prema primarnoj memoriji. Obično je izvedena kao diskovna ili vrpčasta memorija. Kako bi se mnoštvo podataka moglo efikasno obrađivati i brzo pronalaziti, trebaju se na računalu na najprikladniji način organizirano pohraniti na perifernoj, tzv. sekundarnoj memoriji. Imamo organizaciju podataka koja je hijerarhijska, najviši nivo predstavlja baza podataka, a niži elementi su datoteka, slog, polje, znak i bit.

Cilj već naveden, tj. problem latentnosti memorije i kapaciteta memorije rješava se upotrebom **memorijske hijerarhije** i određenim tehnikama. **Tehnika preklapanja** (engl. interleaving) primjenjuje se na hijerarhijskoj razini glavne memorije i dopušta istodobni pristup većem broju memorijskih modula. Na taj se način povećava brzina pristupa memoriji te se povećava i propusnost memorije. Posebnom organizacijom nezavisnih memorijskih modula mogu se ublažiti posljedice sporosti memorije, a to se izvodi tako da se slijedne memorijske lokacije nalaze u različitim memorijskim modulima. Kao primjer „u računalnom sustavu CDC 6600 slijednim adresama naslovljavaju se memorijske lokacije koje se nalaze sa različitim memorijskim modulima, odnosno ako su adrese slijedne naslovit će se sve 32 banke prije nego se ponovo referencira prva memorijska banka. Budući da su banke neovisne, dolazi do istodobnih aktivnosti memorijskih banaka i preklapanja u njihovu pristupu. Na taj se način postiže povećanje brzine rada memorije.“ (prema Ribarić, S., str. 51.).

Djelotvorno rukovanje memorijom na razini **glavna memorija – priručna memorija** dopušta pristup instrukciji ili podatku u vremenu koje odgovara bržoj, tj. priručnoj memoriji.

Uporaba koncepta **virtualne memorije** na razini **glavna memorija – sekundarna memorija** stvara korisniku privid o gotovo neograničenom kapacitetu memorije (tj. točnije, o glavnoj memoriji koja je kapaciteta jednakog sekundarnoj memoriji) i brzine jednake glavnoj memoriji.

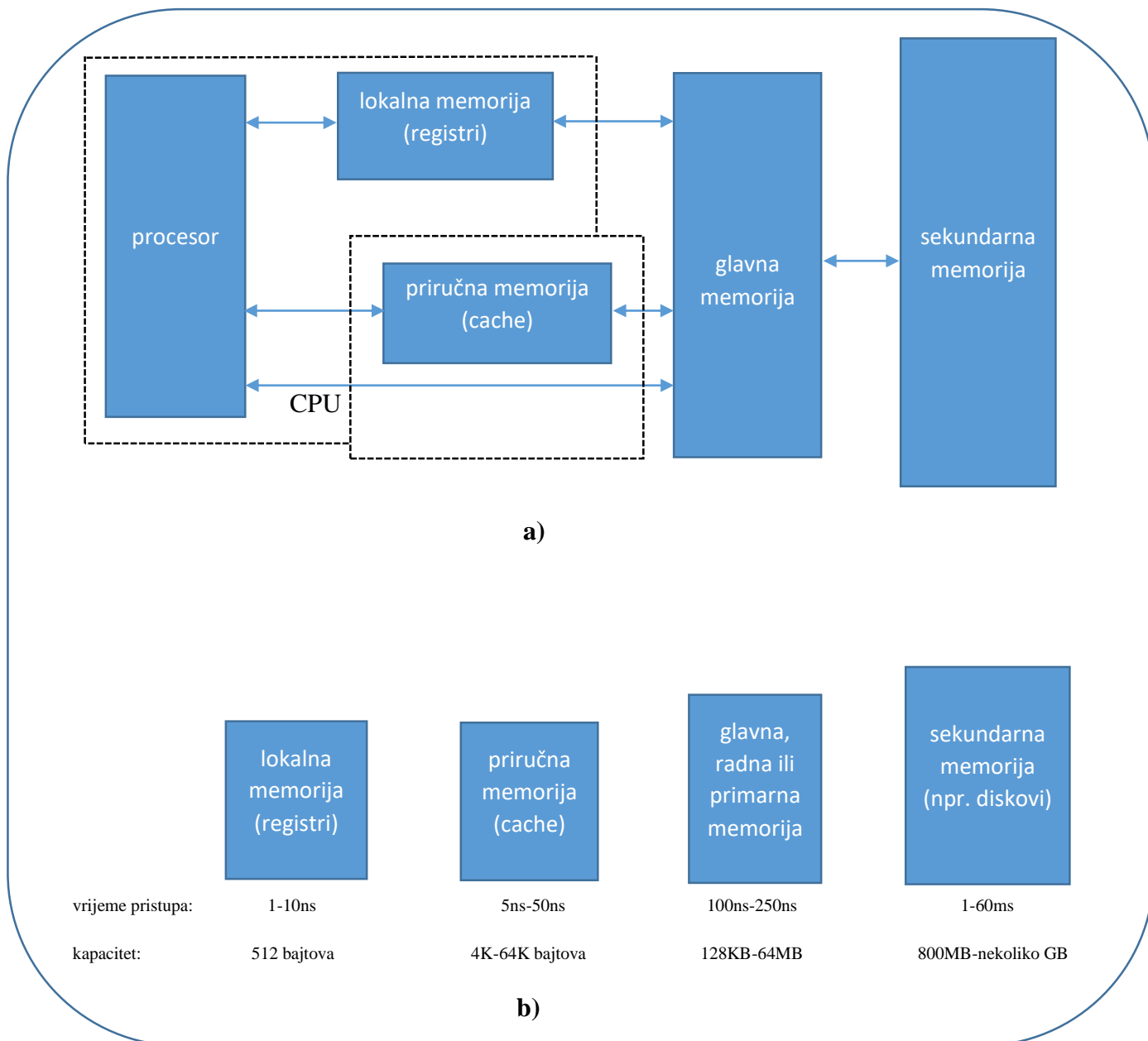
Upotreba virtualnog memorijskog sustava podrazumijeva razlikovanje logičkog adresnog prostora i fizičkog adresnog prostora, a djelotvorno preslikavanje postiže se dijeljenjem tih

dvaju prostora u blokove. Zbog toga, u funkcionalnom smislu glavna i središnja komponenta ovog koncepta je mehanizam adresnog preslikavanja kojim se veliki logički adresni prostor preslikava u puno manji fizički adresni prostor. U logičkom adresnom prostoru blokovi stalne veličine zovu se stranicama. U fizičkom adresnom prostoru blokovi stalne veličine zovu se straničnim okvirima.

**Virtualni memorijski sustav sa straničenjem** naziva se takav sustav koji posjeduje stalne veličine blokova, a **virtualni memorijski sustav sa segmentacijom** naziva se onaj memorijski sustav, koji u svom mehanizmu adresnog preslikavanja upotrebljava blokove promjenjive veličine (segmenti).

Imamo još i **virtualni memorijski sustav sa straničnim segmentima** ili **segmentacije sa straničenjem** (engl. paged segmentation; segmentation with paging), koji se dobiva kombinacijom straničenja i segmentacije, gdje su segmenti podijeljeni u stranice.

Jedan od pokazatelja djelotvornosti virtualnog memorijskog sustava je **omjer pogotka**, koji je vrlo visok zahvaljujući **prostornoj** i **vremenskoj lokalnosti** programa. Vremenska lokalnost ukazuje na lokalnost u vremenu u našoj bliskoj budućnosti, ona ukazuje da će se određeni program obraćati onoj programskoj sekvenci i onim podacima kojima se obraćao u bliskoj prošlosti. Prostorna lokalnost ukazuje na lokalnost u prostoru, tj. u bliskoj budućnosti program će se obraćati onim programima i podacima čije su adrese bliske onima kojima se zadnji put obraćao. Virtualna memorija stvara korisniku privid da je veličina i brzina glavne memorije jednaka kapacitetu i brzini sekundarne memorije.



**Slika 1. Memorijska hijerarhija**

Izvor: „modificirano prema Ribarić, S., str. 262.“

Gornja slika prikazuje memorijsku hijerarhiju. Vidi se da glavna ili primarna memorija pojavljuje se kao memorija koja ima kapacitet sekundarne memorije, odnosno vrlo veliki kapacitet GB, a brzinu jednaku brzini najbrže memorije u hijerarhiji, tj. brzini centralne procesne jedinice. Lokalna memorija ili registri imaju istu brzinu kao sklopovi procesora. Priručna ili cache memorija služi za ubrzanje rada procesora, nalazi se između centralne

procesne jedinice (CPU) i glavne memorije. Sekundarna memorija je dosta spora u odnosu na primarnu memoriju, ali ima vrlo veliki kapacitet reda GB.

## 1.2. LOGIČKA ORGANIZACIJA MEMORIJE

Suvremeni informacijski sustav predstavlja organizirani sustav ljudi, tehnologije i postupaka za prihvaćanje odnosno prikupljanje, obradu, čuvanje i komunikaciju podataka i informacija.

Moderni računalni sustavi mogu služiti za razne namjene, ovisno o zadacima i funkcijama koje sustav mora obaviti te će i njegova organizacija biti vrlo različita. Međutim, takvi sustavi u ovom ili onom obliku moraju obavljati neke osnovne funkcije. Memorijski sustav suvremenih računalnih sustava organiziran je kao hijerarhijski sustav u više razina. **Prvu razinu** memorijskog sustava nazivamo **primarnu memoriju**, a nazivamo je još i središnja memorija (engl. central storage). **Drugu razinu** memorijskog sustava nazivamo **proširena središnja memorija** (engl. extended core storage), a ostvarena je s magnetskim jezgicama. **Treću razinu** nazivamo **pravom sekundarnom memorijom**, koja je ostvarena magnetskim bubnjevima ili diskovima.

Prema Ribarić, S., str. 51. „Primarna memorija računalnog sustava CDC 6600 organizirana je u 32 nezavisne memorijske banke ili memorijska modula. Memorijski modul je osnovni građevni blok memorijske jedinice i ima kapacitet 4096 60-bitnih riječi. U modulu se može istovremeno pristupiti samo jednoj riječi. Vrijeme pristupa, točnije memorijski ciklus (ukupno vrijeme pristupa i vrijeme potrebno za obnavljanje podatka) znatno je dulje negoli vrijeme koje je potrebno za sam prijenos podataka i izvođenje osnovnih operacija u centralnoj procesnoj jedinici. Komunikacija s memorijom zato u računalnim sustavima predstavlja usko grlo obrade.“

Ovaj problem sporosti memorije rješava se posebnom organizacijom nezavisnih memorijskih modula te njihovom organizacijom kao **memorije s preklapanjem** (engl. interleaving), gdje se slijedne memorijske lokacije nalaze u različitim memorijskim modulima.

Prema Ribarić, S., str. 51. „Slijednim se adresama naslovljavaju memorijske lokacije koje se nalaze u različitim memorijskim modulima. Ako su adrese slijedne, naslovit će se sve 32 banke prije negoli se ponovno referencira prva memorijska banka. Budući da su banke neovisne, dolazi do istodobnih aktivnosti memorijskih banaka i preklapanja u njihovu pristupu. Na taj se način postiže povećanje brzine rada memorije.“



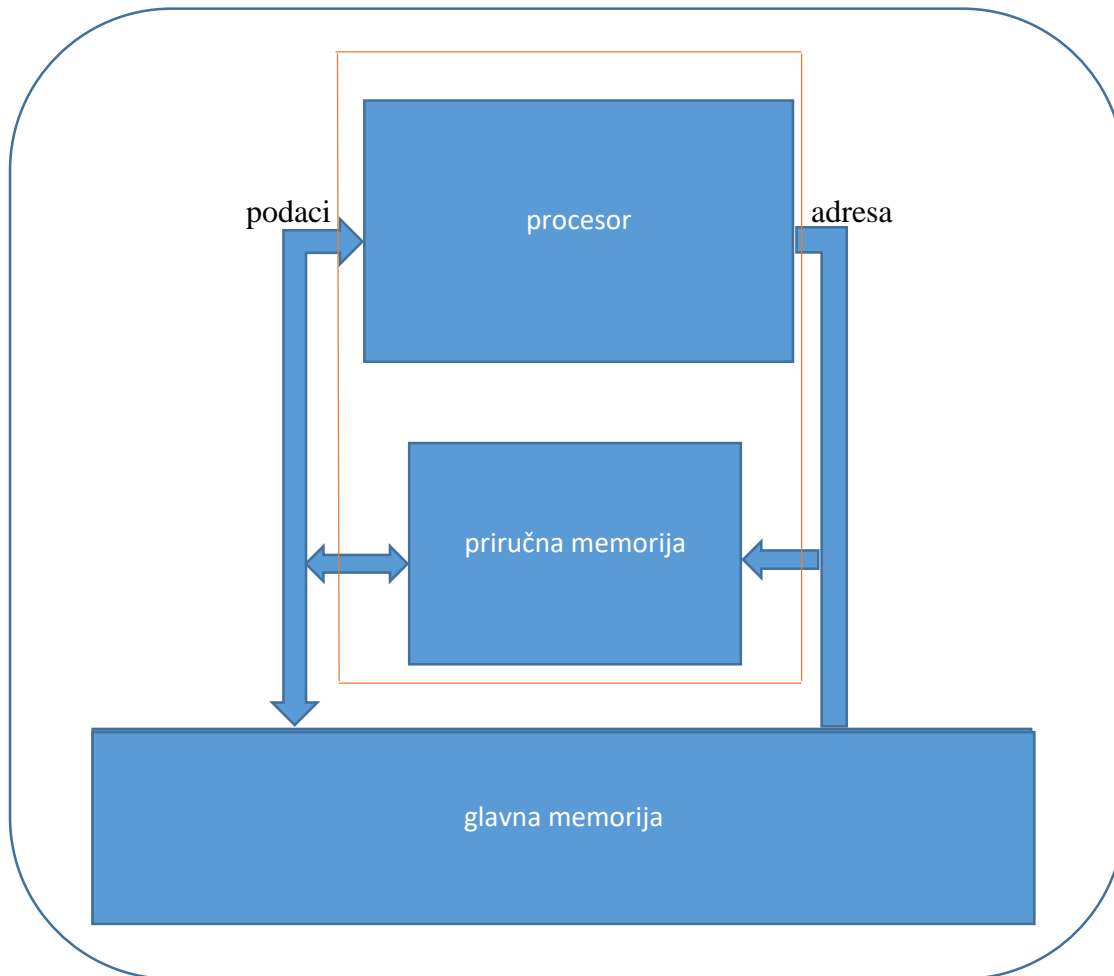
Pod memorijskim modulom podrazumijeva se klasičan memorijski modul von Neumannova računala, gdje su slijedne adrese jednoliko razdijeljene po modulima. Kada se zatraži određeni podatak (instrukcija) na određenoj adresi „a“ istodobno se pristupa i lokacijama s adresom  $(a+1, a+2, \dots, a+N - 1)$ , te se svi ovi sadržaji istovremeno pribavljaju u posebne registre procesora koji oblikuju spremnik.

Prema Ribarić, S., str. 232. „Budući da slijednost adresiranih riječi ovisi i o strukturi programa (razmještaja instrukcija i podataka) i o pojavljivanju upravljačkih instrukcija (instrukcije uvjetnog i bezuvjetnog grananja, instrukcije za pozivanje potprograma) faktor raspršenosti adresa može biti veći od  $N$ . U tom slučaju postoji rješenje koje se sastoji u upotrebi male, ali vrlo brze **priručne memorije** (engl. cache, local memory) koja se smješta između radne memorije i procesora.“

Upotrebom memorije s preklapanjem, zatim pribavljanjem instrukcija i upotrebom priručne memorije dolazi do povećanja propusnosti memorije računalnog sustava. Na ovaj način smanjuje se vrijeme obrade u protočnom segmentu koji pribavlja instrukcije, te se zbog toga daje velika važnost protočnoj organizaciji procesora. Složenije izvedbe memorije s preklapanjem izvode se na taj način da se upotrebljava poseban sklop tzv. **memorijski upravljački sklop** (engl. memory controller) i on ima ulogu inteligentnog posrednika između procesora i memorije, a memorija je organizirana u  $N$  nezavisnih memorijskih modula.

Priručna memorija spada u brze memorije kapaciteta od 4K do 64K ili više stotine KB, gdje je odnos brzine priručne memorije i glavne memorije 5:1 ili više. Ona izravno podržava rad procesora te glavna memorija pokazuje da djeluje brzinom priručne memorije te ona pohranjuje tekuće aktivne segmente programa i podataka.

Za uspješan rad cijele memorijske hijerarhije veoma je važno svojstvo **lokalnosti**, koja se izražava radnim skupom  $W$  (engl. working set), a on predstavlja skup memorijskih lokacija ili blokova koji u vremenu  $t$  su naslovljavani u posljednjih  $h$  pozivanja.



**Slika 2. Putovi podataka između glavne memorije, priručne memorije i procesora**

Izvor: „modificirano prema Ribarić, S., str. 266.“

Gornja slika prikazuje odnos glavne memorije, priručne memorije i procesora, te vidi se da postoje putovi kolanja podataka između glavne memorije i procesora, glavne memorije i priručne memorije i priručne memorije i procesora.

Priručna memorija veoma je važna u povećanju performanse cijelog računalnog sustava, jer ima važnu ulogu u mehanizmima za ubrzanje rada procesora. Ona predstavlja jednu iznimno brzu komponentu, a nalazi se između glavne memorije i centralne procesne jedinice. Vrijeme pristupa priručnoj memoriji je kraće nego vrijeme pristupa glavnoj memoriji, te je veoma važno potrebne podatke koje najčešće upotrebljavamo držati u priručnoj memoriji.

## 2. OPERACIJSKI SUSTAV I NJEGOVE FUNKCIJE

Sklopovlje računala je sastavljeno od nekoliko komponenata, kao što je procesor, radni spremnik, vanjski spremnik te ostalih raznih ulazno – izlaznih naprava. Korisnici računala pomoću različitih programa mijenjaju cijeli računalni sustav u jedan virtualni stroj. Skup osnovnih programa koji imaju zadatak provođenje mnogobrojnih radnih zahvata, tj. izvođenje operacija računala naziva se operacijski sustav.

Prema Kiš, M., str. 254. operacijski sustav (engl. operating system) je „program koji povezuje sklopovske dijelove računala (hardware) s programskom opremom (software) i omogućuje korisniku jednostavnu uporabu računala pomoću naredaba operacijskog sustava. Operacijski sustav provjerava i inicijalizira sve logičke cjeline računala i vanjske uređaje (engl. external device) te učitava programe koji upravljaju tim cjelinama (engl. driver).“

Operacijski sustav ima određene zadatke koje mora izvršiti, a to su olakšanje uporabe računala i organizacija djelotvornog iskorištavanja svih dijelova računala. Računalo može obavljati više zadataka tako da se operacijski sustav brine da se procesor prebacuje s izvođenja jednog niza instrukcija na drugi niz instrukcija i tako podržava višeprogramski rad. Svakom programu operacijski sustav mora omogućiti pristup do potrebnih datoteka i svih ostalih sredstava da tako budu najbolje iskorištena. Daljnji zadatak operacijskog sustava je da mora omogućiti ostvarenje komunikacije između računala. Sve operacije koje operacijski sustav pokreće moraju se izvršiti. Određene zahtjeve za pokretanje operacija na računalu daje korisnik računala (čovjek može zahtijevati da se određeni program pokrene na računalu, da se neka datoteka premjesti s diskete na disk i obrnuto, pretražuje se imenik datoteka itd.), a neki zahtjevi dolaze neposredno iz programa (zahtjevi za obavljanje ulazno – izlaznih operacija). Utvrđeni način postavljanja zahtjeva operacijskom sustavu i sam izgled povratnih poruka operacijskog sustava mora biti dogovoren, što takvu komunikaciju zovemo sučeljem. Sučelje se koristi za dogovoreni način uspostavljanja veze među nekih razdvojenih cjelina.

Operacije operacijskog sustava može pokrenuti čovjek kroz korisničko sučelje ili program preko sučelja primjenskog programa. Programi i svi podaci trajno se čuvaju u obliku datoteka u vanjskim spremnicima računala. Zbog toga je za svaki operacijski sustav najbitnije upoznati rad s datotekama, budući da se sva događanja na računalu odnose na premještanje, mijenjanje i pohranjivanje datoteka. Odgovarajući programi pohranjeni su u obliku datoteka. Operacije koje omogućava operacijski sustav vrlo su složene.

Prema Budin, L., i suradnici str. 3. primjer postupka pripreme jednog programa i njegova korištenja odvija se u nekoliko faza: „Najprije se iz datoteke aktivira program za uređivanje teksta. On prihvaća znakove s tipkovnice i pohranjuje ih postupno u datoteku izvornog programa. Nakon toga se iz svoje datoteke aktivira kompilator. On prihvaća datoteku izvornog teksta kao ulaznu datoteku i uz dodatno korištenje bibliotečne datoteke izgrađuje strojni oblik programa, koji se opet pohranjuje u datoteku. Novostvoreni strojni program se aktivira i postaje program koji prihvaća ulazne podatke s tipkovnica ili iz unaprijed pripremljene podatkovne datoteke. Taj program izračunava rezultate i pohranjuje ih u datoteku rezultata. Na kraju, može se aktivirati neki program za crtanje koji će kao ulaznu datoteku prihvatiti datoteku rezultata i proizvesti crtež na papiru i rastersku sliku na zaslonu. Iz ovog je prikaza vidljivo kako se računalo pretvara iz jednog virtualnog stroja u drugi jednostavnom zamjenom primjenskih programa. Operacijski sustav mora omogućiti da se te pretvorbe obavljaju što jednostavnije.“

Hijerarhijska struktura računalnog sustava predstavlja slojevitú strukturu sustava. Osnovnu razinu računalnog sustava predstavlja računalno sklopovlje, na koje je oslonjena razina operacijskog sustava. Na razinu operacijskog sustava oslanja se razina primjenskih programa, gdje korisnici kroz sučelja programa gledaju kako se računalni sustav transformira u virtualni stroj. Svi programi bez obzira u kojem su programskom jeziku pripremljeni, trebaju biti prevedeni u strojni oblik prije nego li započne njihovo izvođenje. Strojni oblik programa sastoji se od niza strojnih instrukcija, a one u sklopovlju računala čine određene osnovne operacije.

Funkcije operacijskog sustava oslonjene su na sklopovlje i ostvarene su nizom strojnih instrukcija. Korisnici moraju dobro poznavati svojstva i mogućnosti koje pruža određeni operacijski sustav, te moraju poznavati osnovne mehanizme kojima se realiziraju pojedine funkcije operacijskog sustava.

Operacijski sustav ima zadatak da vodi evidenciju o stanju memorije računalnog sustava. On mora u svakom trenutku znati koji su okviri zauzeti, koji procesi koriste pojedine okvire te koji su okviri trenutno slobodni u memoriji. U sustavnom adresnom prostoru čuvaju se u tablicama sve informacije važne za odvijanje procesa, tako se za svaki proces oblikuje **procesni informacijski blok** ili **kontrolni blok** (engl. process information block, process control block). Također sve informacije vezane za gospodarenje memorijskim prostorom pohranjivat će se u procesni informacijski blok, tu se nalaze spomenute tablice sektora gdje je određeni program procesa pohranjen na disku, ovdje nalazimo podatke o smještanju programa u radnom spremniku te ostvarenje virtualnog spremnika.

U radnom spremniku može se držati istodobno više programa, koji su smješteni u razne dijelove spremnika. Kad govorimo o gospodarenju memorijskim prostorom kroz povijesno razdoblje, vidimo da se dodjela spremnika zasnivala na podjeli spremnika na dijelove stalne veličine koje nazivamo **stalne particije** (engl. fixed partitions). Particije mogu biti jednake ili različite veličine, a particije različitih veličina bolje iskorištavaju radni spremnik.

Suvremene informacijsko – komunikacijske tehnologije imaju takve računalne sustave koji upotrebljavaju procesore s takvim spremničkim međusklopovima, koji omogućuju da operacijski sustavi djelotvorno gospodare spremničkim prostorom. Dio operacijskog sustava koji ima zadatak gospodarenja memorijskog spremničkog prostora mora osposobiti da se pojedini okviri fizičkog adresnog prostora dodjeljuju određenom procesu.

Prema Budin, L. i suradnici, str. 230. „Dobro gospodarenje spremničkim prostorom zahtijeva poznavanje trenutačnog stanja svakog pojedinačnog okvira fizičkog radnog spremnika. Najprikladnije je sve potrebne podatke o okviru smjestiti u jedan opisnik. Opisnik mora imati i mjesta za kazaljke kako bi se lako mogao premještati iz liste u listu.“

Postupci gospodarenja memorijskim prostorom znatno određuju rad računalnih sustava, u kojima je najčešće gospodarenje memorijom zasnovano na virtualnom spremniku.

### 3. OSNOVE GOSPODARENJA MEMORIJSKIM PROSTOROM

U današnje vrijeme brzog i velikog razvoja informacijsko – komunikacijske tehnologije, računalni sustav spada u jedan od najsloženijih umjetnih sustava kojeg je stvorila čovjekova inteligencija. Suvremena informatička tehnologija ušla je u sva područja ljudskog života i svakodnevno kako ona napreduje tako napreduju i korisnički zahtjevi. Korisnički programi i mnogobrojne aplikacije danas su sve složeniji te se u računalnim sustavima zahtijeva da posjeduju sve više memorije. Ona je jedna od osnovnih komponenata suvremenog računalnog sustava i jedan vrlo važan resurs, kojim se mora pažljivo i inteligentno upravljati, a predstavlja jedno veliko polje okteta ili riječi koji svaki imaju svoju adresu. Iz memorije procesor dohvaća naredbe u ovisnosti o vrijednosti upisanoj u programsko brojilo, a krajnji traženi rezultati ovih naredbi mogu biti dodatni pristupi memoriji kako bi se dohvatilo operande tj. pohranilo rezultate operacija.

U današnje vrijeme prosječna kućna računala posjeduju oko tisuću puta više memorije nego IBM 7094, koje je bilo napravljeno početkom 1960-tih godina i predstavljalo je onda najveće računalo toga doba na svijetu. U posljednja četiri desetljeća razvoj računalnih sustava postigao je progresivno povećanje složenosti arhitekture, procesori postaju sve složeniji, a to se odražava u velikim skupovima instrukcija (engl. instruction – naredba, instrukcija, uputa; opis djelovanja koje izvodi računalo na sklopovskoj razini) te vrlo složenim instrukcijama te velikom broju raspoloživih načina adresiranja.

Prema Ribarić, S., str. 5. „Instrukcija se može sastojati od najmanje jedne riječi (16 bita) pa sve do jedanaest riječi. Prva riječ u instrukciji zove se **operacijska riječ** i određuje operaciju koja će se izvršiti te duljinu same instrukcije. Preostale riječi instrukcije zovu se **riječi proširenja** i služe za daljnju, detaljniju specifikaciju instrukcije i operande).“

Prema Ribarić, S., str. 17. „Velik skup instrukcija, veličine nekoliko stotina instrukcija, sadrži velik broj složenih, vremenski zahtjevnih instrukcija. One koriste velik broj različitih načina adresiranja i imaju različite formate. Složenost instrukcija ne znači i brže izvođenje programa.“

Većina suvremenih računala tj. procesora (arhitektura CISC ili RISC) temelje se na konceptu von Neumannova računala, kojima je temeljna značajka da im je program u obliku slijeda kodova (instrukcija) pohranjen u memoriji u kojoj su u istom obliku pohranjeni i podaci.

### 3.1. MEMORIJSKO-UPRAVLJAČKA JEDINICA (MMU)

**Memorijsko – upravljačka jedinica** (engl. memory management unit – MMU) je jedinica za upravljanje memorijom, hardverska je to komponenta te predstavlja dio procesora koji prevodi logičke (virtualne) adrese u fizičke. Procesor generira adresu i ona se promatra kao logička adresa. Adrese koje se koriste u izvornom programu jesu simboličke, a nakon prevođenja i povezivanja one postaju logičke adrese i tako čine logički adresni prostor.

Skup adresa koje su dodijeljene memorijskim lokacijama čine fizički adresni prostor i nazivaju se fizičke adrese. U memoriji računalnog sustava to su adrese koje vidi sklop za upravljanje memorijom, te pomoću sklopa počinje princip transformacije adrese.

Logički adresni prostor predstavlja adresni prostor kojeg koristi program prije punjenja u memoriju. Zatim, procesor generira logičnu tj. virtualnu adresu koju šalje MMU-u, a on transformira logičku adresu u fizičku adresu i dalje je prosljeđuje u memoriju. Logički adresni prostor prelazi u fizički nakon punjenja u memoriju.

#### 3.1.1. FIZIČKI I LOGIČKI ADRESNI PROSTOR I VEZIVANJE ADRESA

Kako bi se ostvarilo učinkovito rukovanje memorije, treba znati razlikovati **fizički adresni prostor** i **logički adresni prostor**. Za fizičku memoriju može se reći da je oblikuje skup stvarnih, fizičkih memorijskih lokacija glavne memorije na kojima se pohranjuju podaci i programi. Može se promatrati i kao memorija koja je priključena na sabirnicu procesora tj. računala. **Fizički adresni prostor** predstavlja skup adresa koje se jednoznačno dodjeljuju tim memorijskim lokacijama.

Za **logički adresni prostor** može se reći da je to skup logičkih adresa. **Logičku adresu** može se definirati kao adresu koju generira program ili upotrebljava programer kao najmanju programsku jedinicu. Fizički i logički adresni prostor jesu međusobno neovisni prostori, zato skup logičkih adresa koje generira procesor ili program ne ovisi o stvarnoj veličini raspoložive fizičke memorije.

Odnos između veličine logičkog i fizičkog prostora može biti **logički prostor** koji je **jednak** fizičkom adresnom prostoru (mikroračunalni sustavi koji se temelje na 8-bitnim mikroprocesorima, logički i adresni prostori svaki su veličine po 64 KB).

Logički adresni prostor **manji** je od fizičkog. Upotrebom memorijske banke ili indeksnim preslikavanjem povećava se fizički prostor, ali procesor još uvijek je ograničen logičkom adresnom prostoru.

Logički adresni prostor **veći** je od fizičkog. Nastali problem rukovanja memorijom mijenja se pojavom 16-bitnih, 32-bitnih i 64-bitnih mikroprocesora. U današnje vrijeme svjesni smo potrebe velikog adresnog prostora, arhitekti neprestano povećavaju širinu adresne sabirnice procesora – 16 M, 48 M ili 4 G i više bajtova.

Budući da postoji nesklad u veličini fizičkog i logičkog adresnog prostora, u računalni sustav ugrađuju se mehanizmi koji se ostvaruju upotrebom sklopovlja i sustavne programske opreme. Kada je logički adresni prostor veći od fizičkog primjenjuje se koncept **virtualne memorije**. Virtualna memorija premošćuje jaz u veličini između glavne memorije i sekundarne memorije. Ona podrazumijeva takvu strategiju dodjele memorije koja dozvoljava da samo dio programa koji se trenutno izvodi bude smješten u radnoj memoriji računala. Prednost ovakve memorije ogleda se u tome što program može biti i veći od radne memorije, odnosno veličina programa nije ograničena veličinom radne memorije. Programer može napisati određeni program u računalnom sustavu kao da sam sustav raspolaže s neograničenom memorijom, tj. programer sa virtualnom memorijom raspolaže s neograničenim virtualnim logičkim adresnim prostorom. Prednosti virtualne memorije jesu što se korisnički program može izvoditi sa znatno manjom dodijeljenom fizičkom memorijom, a tako se povećava iskoristivost sustava i propusna moć sustava. Virtualna memorija realizira se kao straničenje na zahtjev (engl. demand paging).

### 3.1.2. ORGANIZACIJA LOGIČKOG ADRESNOG PROSTORA

Svaki programer radi programe u kojima se pomoću prevodioca dalje generiraju logičke adrese. Zbog toga struktura i organizacija toga **logičkog adresnog prostora** postaje veoma važna i predstavljena je određenim modelima.

**Linearni memorijski model** ili **jednodimenzionalni** (engl. linear memory, flat model, unsegmented model) je najjednostavnija organizacija logičkog adresnog prostora. Logički adresni prostor sastoji se od adrese od 0 do neke maksimalne vrijednosti. U ovoj organizaciji izravno se ne razlikuju određeni podaci i programi te je cijeli prostor jedan **homogeni prostor** (npr. sa 24-bitnom logičkom adresom pristupa se linearnom logičkom prostoru veličine 16 M bajtova).



**Model memorije u odsječcima** (engl. segmented model) je logički adresni prostor kojeg čine brojni linearni odsječci ili segmenti. Izdvojeni linearni logički prostor predstavlja svaki odsječak. Ovim modelom organizacije ostvaruje se pristup podacima upotrebom segmentnog broja ili imena segmenta i pomaka unutar segmenta. Ovaj model memorije naziva se i **dvodimenzionalni model**, budući da u oblikovanju adrese sudjeluju dva elementa (logičku adresu čine dva dijela: 4-bitnog segmentnog broja i 20-bitnog pomaka).

### 3.1.3. ALOKACIJA MEMORIJE – DODJELA MEMORIJE PROCESIMA

Kako bi se osigurao nesmetan rad računalnog sistema, neophodno je potrebno što efikasnije dodijeliti različite dijelove memorije. Zbog toga, glavni problem koji nastaje pri upravljanju memorijom je dodjela slobodne memorije procesima a to se naziva alokacija memorije.

Najčešće se memorija dijeli na dvije particije (raspodjela, podjela). To je područje u radnoj memoriji koje se dodjeljuje programu dok traje njegovo izvođenje. **Jedna particija** namijenjena je rezidentnom dijelu operacijskog sustava (engl. kernel space) i to je najniži dio operacijskog sustava. To je skup procedura koje se nalaze u računalu zbog prosljeđivanja podataka između pojedinih njegovih dijelova koji izvode najjednostavnije operacije kao što je upravljanje memorijom (engl. memory management), dodjela sredstava (engl. resource allocation) ili nadzor ulaza i izlaza podataka (input/output). **Druga particija** namijenjena je višim dijelovima operacijskog sustava odnosno korisničkim procesima (engl. user space).

U najnižem dijelu memorije nalazi se tabela prekidnih rutina, što označava postupak povezivanja dvaju korisnika prema određenoj adresnoj informaciji koja se nalazi u zahtjevu za prijenos. Pri upravljanju tj. gospodarenjem memorijskim prostorom nastaje jedan glavni problem, a to je dodjela slobodne memorije procesima koji su u ulaznom redu (alokacija memorije).

Tehnike za dodjelu memorije procesima mogu se podijeliti na **kontinualnu alokaciju** (engl. contiguous allocation), gdje se fizički i logički adresni prostor procesa sastoji od kontinualnih niza memorijskih riječi, a da pri tome memorijske particije po veličini mogu biti jednake ili različite. **Diskontinualna alokacija** (engl. discontinuous allocation) je fizički adresni prostor procesa koji nije realiziran kao kontinualan niz memorijskih adresa, već on obuhvaća straničenja, segmentacije i straničenja sa segmentacijom.

### 3.2. GLAVNE FUNKCIJE I ZAHTJEVI SUSTAVA ZA GOSPODARENJE MEMORIJOM

U glavne funkcije sustava za gospodarenje memorijom spada vođenje evidencije o raspoloživosti slobodnih dijelova memorije, alokacija tj. dodjela memorije procesima, oslobađanje memorije kada nije više potrebna procesima, te zamjena sadržaja između glavne memorije i diska kada glavna memorija nije dovoljno velika za dani proces.

Središnju komponentu računalnog sustava čini glavna, radna ili primarna memorija. Međutim, u današnjem brzom razvoju informacijske tehnologije pred informatičkim stručnjacima postavljaju se veliki zahtjevi u pogledu kapaciteta glavne memorije. Ovaj sve veći problem rješava se upotrebom viših programskih jezika te razvojem sve složenijih programskih sustava u višekorisničkim okruženjima. Glavne memorije današnjih računalnih sustava imaju kapacitete 8,16 i više MB (engl. megabyte - mjerna jedinica za kapacitet memorije, 1MB iznosi 1 048 576 okteta /byte/ ili 8 388 608 bitova) a mnoge primjene zahtijevaju sve veći kapacitet memorije (nekoliko stotina MB ili nekoliko GB).

Memorija se sastoji od velikog niza različitih riječi ili bajtova od kojih svaki posjeduje svoju vlastitu adresu. Iz memorije procesor dobavlja instrukcije, s obzirom na vrijednost programskog brojača, a ove instrukcije mogu izazvati dodatno učitavanje i pohranjivanje na posebne memorijske adrese.

U jednom ciklusu instrukcijskog izvršavanja prvo se dobavlja instrukcija iz memorije. Nakon toga slijedi njeno dekodiranje, koje podrazumijeva dohvaćanje operanda iz memorije računala. Zatim slijedi izvršavanje instrukcija nad operandima, a rezultati mogu se pohraniti u memoriju. Memorijska jedinica vidi samo niz memorijskih adresa, a da pri tome ne zna kako su ove adrese generirane (da li pomoću instrukcijskog brojača, indeksiranja ili dr.), niti za što su namijenjene (ili za instrukcije ili za podatke).

Jedini memorijski uređaji kojima procesor direktno pristupa su radna memorija i registri procesora. Zadatak većine procesora je da može dekodirati instrukcije i tako izvesti jednostavne operacije nad sadržajem registara. Ove operacije može izvesti brzinom od jedne do nekoliko operacija po jednom radnom taktu. Kod radne memorije se pristupa putem memorijske sabirnice (prema Kiš, M., str. 45. engl. „bus – skup vodiča koji služe za jednosmjerni ili dvosmjerni prijenos podataka, adresa i upravljačkih signala između mikroprocesora i ostalih dijelova računalnog sustava – vanjske sabirnice; ili u mikroprocesoru između njegovih sastavnih dijelova – unutarnje sabirnice; skup vodiča kojima se povezuju sastavni dijelovi računala; put prijenosa podataka“), a samo pristupanje može trajati i nekoliko radnih taktova

procesora. U ovom slučaju procesor miruje, budući da nema potrebne zatražene podatke za dovršenje instrukcije, koja se u međuvremenu izvršava. Ovaj problem mirovanja procesora je prisutan ali nedopustiv, te se kao rješenje dodaje brza memorija između procesora i glavne memorije. Svojevrsan amortizer koji se koristi između različitih brzina procesora i glavne memorije naziva se keš memorija (prema Kiš, M., str. 47. engl. “cache – predmemorija, brza memorija, priručna memorija, skrivena memorija – to je posebna memorija malog kapaciteta i brzog pristupa podacima kojom upravlja za to predviđen sklop (cache controller); memorija za pamćenje dijela podataka ili dijelova programa koji su potrebni u izvođenju iduće operacije“).

Može se sagledati da sa aspekta upravljanja – gospodarenja memorijom tj. memorijskim prostorom, informatički stručnjaci susreću se kod suvremenih računalnih sustava sa navedenim problemima dijeljenja memorije i usklađivanja brzina procesora i glavne memorije. Također, sve više postaje prisutno i veoma važno pitanje zaštite, jer se moraju osigurati mehanizmi zaštite memorije i tako priječiti da korisnički programi modificiraju kod ili strukturu podataka (namjerno ili slučajno) drugih korisničkih procesa ili procesa operativnog sustava.

## 4. PROBLEMI UPRAVLJANJA MEMORIJOM

Korisnički programi i sve veći broj aplikacija, u današnje vrijeme brzog razvoja informatičke tehnologije, postaju sve više složeniji, veći i sve više zahtjevniji te sve više nedostaje količine memorije koja je dodijeljena programu. Sve više se informatički stručnjaci bave pitanjem kako povećati količinu memorije u računalnim sustavima i to postaje osnovno pitanje informacijskog napretka. Količina podataka koja se može pohraniti u memoriju postaje sve veća, te računalni sustavi vape za proširenjem memorijskog kapaciteta i ubrzanja pohrane.

Programeri htjeli bi imati na raspolaganju beskonačno brzu i veliku memoriju, koja prilikom uskraćivanja napajanja ne gubi svoj sadržaj, a jedan od zahtjeva je da ne bude skupa kako bi većoj masi bila dostupna. Pitanje veličine memorije postaje za suvremene stručnjake sudbonosno pri opstanku na tržištu i traže se mogućnosti kako pružiti korisnicima takvu veliku memoriju na računalnim sustavima. Danas je to riješeno tako što suvremena računala imaju podijeljenu memoriju prema hijerarhiji, a operacijski sustav ima zadatak da koordinira upotrebu ovih memorija.

Zbog toga, imamo različite realizacije raspoređivanja procesa, koje omogućavaju povećanje performansi sustava, a svaki od njih podrazumijeva držanje više procesa u memoriji računala. To znači da se ukupna memorija mora dijeliti na dva dijela: jedan dio je za jezgru operacijskog sustava, a drugi dio je za programe koji se trenutno izvršavaju.

Jedan od osnovnih zadataka operacijskog sustava je dijeljenje memorije, a on se ogleda u pravilnom upravljanju memorijom. U tu svrhu postoje različite metode kojima se ostvaruje upravljanje memorijom. Sam izbor metode upravljanja memorijom računalnog sustava zavisi od bezbroj faktora, a najviše od hardvera.

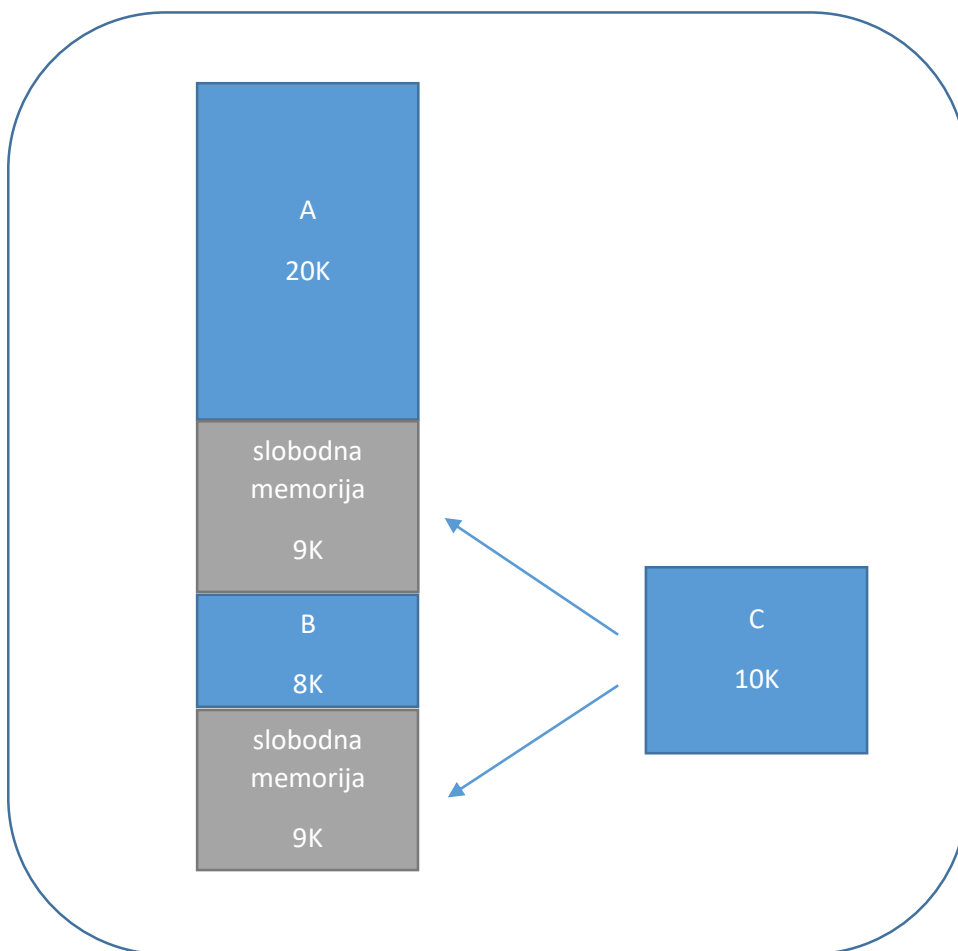
### 4.1. UNUTARNJA I VANJSKA FRAGMENTACIJA

Segmentacija predstavlja osnovni temelj virtualnog memorijskog sustava, izvire izravno iz programskih jezika i strukturiranog programiranja, a prvi puta je komercijalno primijenjena u računalnom sustavu Burroughs 5500 1973. godine. Sa gledišta jezično usmjerenih arhitektura, segmentacija ima uspješan koncept, ali ma i svoj veliki nedostatak. Nedostatak se sastoji u tome što je prisutan složen postupak zamjene i smještanja segmenata radi njihove promjenjive duljine, a sve to dovodi do **fragmentacije memorije**, koju često nazivamo i **vanjska fragmentacija memorije**. Problem fragmentacije može postati ozbiljan proces, koji znatno umanjuje učinkovitost cijelog računalnog sustava.

Budući da se mnogobrojni procesi neprekidno upisuju i izbacuju iz memorije, postaje potreba da se slobodna memorija razbija na manje segmente. Kroz dulji vremenski period pojavljuje se ipak problem da iako je ukupna veličina slobodne memorije dovoljna da prihvati nove procese, to nije moguće izvršiti jer ne postoje dovoljno velika kontinuirana područja u memoriji. U memoriju se ne mogu upisati veći programi, premda je ukupna slobodna memorija dovoljna. Nastali problem vanjske fragmentacije može se riješiti premještanjem procesa i spajanjem slobodnih blokova u jedan kontinuirani blok.

Ova realizacija procesa spajanja jedino je moguće izvesti ukoliko se fizičke adrese generiraju dinamički, odnosno za vrijeme izvođenja programa, kada je potrebno premjestiti samo procese i promijeniti sadržaje baznih spremnika. Ako su adrese ovih procesa generirane za vrijeme pripreme programa za izvođenje ili za vrijeme punjenja, neće biti moguće spajanje.

Osim vanjske fragmentacije memorije, imamo i **unutarnju fragmentaciju memorije**, koju možemo definirati kao razliku između zahtijevane memorije od strane procesa i dodijeljene mu memorije. U usporedbi s vanjskom fragmentacijom, unutarnja fragmentacija je praktički zanemariva.

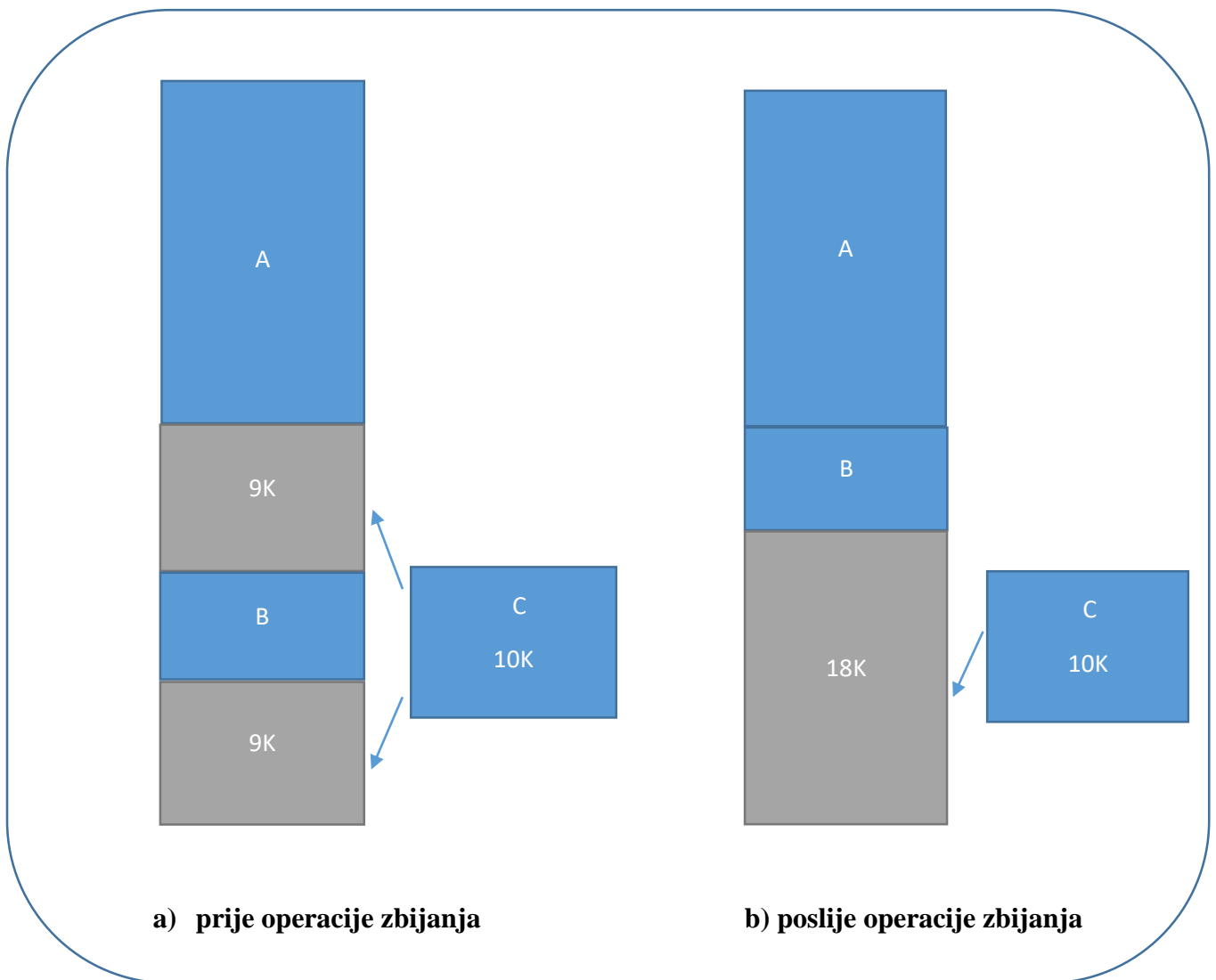


Slika 3. Vanjska fragmentacija memorije

Izvor: „modificirano prema Ribarić, S., str. 322.“

Slika prikazuje vanjsku fragmentaciju memorije, gdje je ukupan slobodan memorijski prostor 18 K, te je veći od novog segmenta C koji iznosi 10 K. Sustav za rukovanje memorijom ima zadatak održavati razinu vanjske fragmentacije što je moguće nižom. Vidi se da segment C ne može smjestiti u glavnu memoriju.

Iz memorije se neprestano upisuju i izbacuju određeni procesi. Premda se slobodna memorija razbija na manje segmente, pojavljuje se problem sa vremenom. Iako je dovoljna ukupna veličina slobodne memorije da prihvati nove procese, to postaje problem, jer ne postoje dovoljno velika kontinuirana područja. Ovakvu nastalu situaciju zovemo vanjska fragmentacija memorije, a ona može predstavljati ozbiljan problem, koji značajno umanjuje učinkovitost cijelog računalnog sustava.



Slika 4. **Primjer operacije zbijanja**

Izvor: „modificirano prema Ribarić, S., str. 323.“

Gornja slika prikazuje primjer operacije zbijanja (engl. compaction, garbage collection), koji rješava problem smještanja novog segmenta C u glavnu memoriju. Memorija na taj način postaje kompaktnija, premda je to vremenski zahtjevna operacija za veliki kapacitet glavne memorije u računalnom sustavu.

## 4.2. SLOBODNE MEMORIJSKE PARTICIJE

Glavna karakteristika suvremenih operacijskih sustava je tzv. „višezadaćnost“, budući da računalni sustavi današnjice omogućuju rad više procesa u isto vrijeme. Ako se implementira višeprocetni rad kod sustava s jednom particijom, mora se prilikom promjene konteksta izbacivati prekinuti proces na disk i upisati drugi u radnu memoriju (engl. swapping). Međutim, ovakav pristup zahtijeva veliko vrijeme izmjene konteksta, što postaje neprihvatljivo i usko grlo samog procesa rada.

U današnje vrijeme povećani su kapaciteti glavne memorije u računalnim sustavima, te se našlo moguće rješenje podjele memorije na više dijelova odnosno particija. Broj particija može biti promjenjiv ili konstantan.

Tako je brojem particija određen stupanj višeprogramskog rada sustava, a u svakoj particiji smješta se po jedan korisnički proces. Kada korisnik pokreće svoj program, njega operacijski sustav stavlja na disk, kao proces koji je prihvaćen na obradu, ali još nije dobio pravo da se natječe za korištenje procesora, tj. izvođenje.

U trenutku kada se oslobodi jedna memorijska particija u računalnom sustavu, program se upisuje u nju, te tako postaje proces koji može dobiti pravo korištenja procesora. Nakon završetka obrade, proces se izbacuje iz memorije te se slobodna particija dodjeljuje dalje sljedećem procesu koji čeka na disku. Sam način rada sastoji se u tome da dok jedan proces čeka na ulaz – izlaz, drugi proces može koristiti procesor, te se tako povećava iskorištenost samog procesora.

Postoje tako fiksne particije memorije sa izdvojenim posebnim ulaznim nizovima za svaku particiju, fiksne particije memorije sa jednim ulaznim nizom, te promjenjive memorijske particije. Kod **fiksnih particija memorije sa izdvojenim posebnim ulaznim nizovima za svaku particiju**, određeni zahtjevi smještaju se u red čekanja na određenu particiju. Tako se događa da ulazni niz za veliku particiju memorije ostane prazan, a niz za malu particiju pun. Premda je veliki dio memorije neiskorišten, uočava se da mali zadaci moraju čekati na memoriju. Kod **fiksnih particija memorije sa jednim ulaznim nizom** vidimo da se određeni zahtjevi procesa smještaju u jedinstveni red čekanja. Kada se oslobodi bilo koja particija, ona će biti dodijeljena sljedećem prvom zahtjevu koji čeka prvi na redu. Budući da nije dobro trošiti velike particije memorije za male zadatke, koristi se takva strategija pretraživanja cijelog ulaznog niza kada se oslobodi neka particija. Odabere se najveći zadatak koji može stati u ovu particiju.



Broj, veličina i stanje memorijskih particija, te da li su slobodne ili zauzete, dinamički se mijenjaju sa aktiviranjem i završavanjem procesa. Sve ovo dovodi do povećanja iskorištenosti memorije računalnog sustava.

Operacijski sustav ima zadatak da vodi evidenciju o slobodnoj i zauzetoj memoriji. Postoje dva načina evidentiranja upotrebe memorije, a to su bitmape (zauzimaju puno mjesta, koriste se za veliku memoriju i mnogobrojne procese) i povezane liste (sporo traženje slobodnog mjesta).

**Bitmape** označavaju memoriju u računalu koja je podijeljena u jedinice za dodjeljivanje, a veličine je od nekoliko riječi do nekoliko kilobajta. Svakoj jedinici za dodjeljivanje odgovara jedan bit u bitmapi, koji ima vrijednost 0 ako je jedinica slobodna, odnosno 1 ako je zauzeta ili obratno. Što je manja memorijska jedinica za dodjeljivanje, veća je bitmapa, a ako je jedinica velika, bitmapa će biti manja. Sa bitmapama može se na jednostavan način voditi potrebnu evidenciju memorijskih riječi kod fiksne veličine memorije, budući da veličina bitmape zavisi samo od veličine memorije i veličine memorijske jedinice. Glavni problem nastaje kada sklop za upravljanje memorijom mora pretražiti bitmapu, a to pretraživanje memorije radi pronalaženja niza određene dužine je spora operacija, te predstavlja glavni nedostatak korištenja bitmapa.

**Povezane liste** predstavljaju drugi način evidentiranja memorije, a sastoji se u održavanju povezanih lista dodijeljenih i slobodnih memorijskih segmenata. U povezanim listama elementi sadrže oznaku u kojoj vidimo da li se radi o procesu ili slobodnoj particiji, početnu adresu, veličinu te pokazatelj na idući unos.

#### 4.3. PODRŠKA ZA DINAMIČKO PUNJENJE I VEZIVANJE

U višekorisničkim računalnim sustavima i naprednijih primjena, **postupak prekrivanja** (engl. overlay) je gotovo neizvodiv i nedjelotvoran, dok je u jednostavnim jedнокorisničkim računalnim sustavima on djelotvoran. U jednostavnim sustavima nesklad između zahtjeva programa za memorijom i kapaciteta stvarne, fizičke memorije rješava se postupkom prekrivanja, gdje programer dijeli program na nekoliko neovisnih programskih blokova (modula) i blokova podataka. Tijekom izvođenja programa svi programski blokovi i blokovi podataka nisu istodobno potrebni, već se trajno u memoriji pohranjuje onaj dio programa kojim se upravlja premještanje blokova između glavne i sekundarne memorije. U sekundarnoj memoriji smješteni su svi ostali moduli te premještaju se u glavnu memoriju samo kad su potrebni. Na taj način moduli „prekrivajući“ stare neaktivne module, premještaju se u isto područje glavne memorije na kojem su bili pohranjeni prethodni moduli. Programer je za ovaj način rješavanja problema trošio dodatno vrijeme na dijeljenju programa u neovisne module koji su pogodni za premještanje.

U višeprogramskim i višekorisničkim sustavima upotrebljavaju se posebne **tehnike upravljanja memorijom**, a njene su značajke **dinamičko rukovanje memorijom**, odnosno postupak prekrivanja. Ove tehnike provode se tijekom izvođenja programa, automatski bez udjela programera ili operatera i temelje se na razlikovanju **logičkog adresnog prostora** i **fizičkog adresnog prostora**, adresnom preslikavanju, dinamičkom dodjeljivanju memorije te zaštitnim mehanizmima pristupa.

Adresno preslikavanje znači da se stvarni, fizički memorijski prostor mora dodijeliti programu prije njegova izvođenja. Dodjeljivanje memorijskog prostora u suvremenim računalnim sustavima izvodi dinamički tijekom izvođenja programa, gdje je važno preslikavanje logičkog adresnog prostora u fizički adresni prostor.

Prema Ribarić, S., str. 303. „Osnovni je problem virtualne memorije upravo izvedba mehanizma preslikavanja velikog logičkog adresnog prostora u puno manji fizički adresni prostor, odnosno preslikavanje logičke (ili virtualne) adrese u fizičku adresu.“

Prema Ribarić, S., str. 304. „Preslikavanje označava promašaj (engl. missing-item fault). Program koji je generirao takvu adresu a privremeno se suspendira dok se zahtijevana riječ ne prenese iz sekundarne memorije u glavnu memoriju. Preslikavanje adrese logičkog prostora u adresu fizičkog prostora ostvaruje se pomoću **tablice preslikavanja** (engl. mapping table). Tijekom izvođenja programa tablica sadrži fizičke adrese koje su pridružene odgovarajućim

logičkim adresama. To pridruživanje izvodi operacijski sustav koji je odgovoran i za odgovarajuće promjene u tablici tijekom izvođenja programa. Tablica preslikavanja može se promatrati i kao skup registara.“

Kako bi mehanizam preslikavanja bio uspješan treba se kapacitet tj. veličina tablice preslikavanja smanjiti, a to će se postići ako element u tablici preslikavanja sadrži adresu bloka podataka a ne adrese pojedinačno naslovljenog podatka. Znači, cilj je rješenje problema u dijeljenju logičkog i fizičkog adresnog prostora na blokove, koji se nazivaju stranicama tj. segmentima ako su blokovi promjenjive veličine.

## 5. PROŠIRENJE MEMORIJSKOG PROSTORA

Proširenje memorijskog prostora vrši se relokacijom (defragmentacija i swap radne memorije). Riječ **relokacija** znači premjestiti, odrediti novi položaj u memoriji. Određeni program može se smjestiti počevši od bilo koje memorijske adrese, također i premještati na disk te ponovno smještati od neke nove adrese. Kada korisnički program zatraži određenu količinu memorije, određenim sistemskim pozivima traži se od operacijskog sustava memorija.

U trenutku kada proces zatraži memoriju jezgra operacijskog sustava tzv. „kernel“ (to je najniži dio operacijskog sustava, skup programa u operacijskom sustavu koji izvode osnovne najjednostavnije operacije kao što je upravljanje memorijom, dodjela sredstava ili nadzor ulaza – izlaza podataka; može predstavljati skup procedura koje su u računalima radi prosljeđivanja podataka između pojedinih njihovih dijelova) ima zadatak da učini važećim određeni raspon adresa. Pri prvom pristupu procesa tom rasponu adresa u memoriji, dešava se alokacija memorijskih stranica. Dešava se da je memorija u računalnom sustavu „prebukirana“ te se dozvoljava alokacija memorije procesu, ako ima dovoljno slobodnog prostora u adresnom prostoru tog procesa. Ako se ne provjerava da li ima dovoljno stvarne, fizičke memorije (RAM + swap), može se dogoditi da ukupna količina memorije koja je dodijeljena svim procesima prelazi ukupnu količinu fizičke i swap (mijenjati mjesto, prebacivati, zamjena, izmjena, promjena) memorije. Tzv. „prebukiranje“ memorije ne stvara obično nikakve dodatne probleme na suvremenim računalnim sustavima, već olakšava zahtjeve na količinu slobodne memorije, a naročito swap memorije. Swapping ili prebacivanje znači da se proces može izvoditi samo ako se nalazi u radnoj memoriji računala. Ovdje se procesi višeg prioriteta čuvaju u memoriji računala, dok se procesi nižeg prioriteta upisuju na disk i stavljaju na čekanje. Uglavnom se kod swappinga razmjenjuju manji dijelovi memorije, a veoma rijetko se izmjenjuju cijeli procesi.

Može se zaključiti da kod multiprogramskog režima rada istovremeno je po nekoliko procesa prisutno u memoriji računala, i sam proces unaprijed ne zna gdje će biti napunjen te se zbog toga ne koriste apsolutne memorijske adrese.

Relokacija je potrebna onda kada je neophodno obaviti neki vid translacije adresa, tj. virtualne adrese je potrebno prevesti u realne. Ovaj proces zove se translacija adresa, a ona se izvodi hardverskom jedinicom MMU.

## 6. ZAKLJUČAK

Brzi razvoj informacijsko – komunikacijske tehnologije uzdrmao je sva područja ljudskog života. Danas živimo u tzv. informatičkom dobu, gdje se brzim napretkom informatičke tehnologije proces gospodarenja memorijom drastično poboljšao hardverski i softverski. Značaj uloge gospodarenja memorijskim prostorom postaje neopisivo važno za brzi razvoj cijelog računalnog sustava. U povijesnom razvitku nastali nedostaci koji su se pojavljivali te problemi koji su se time javljali, pokazali su da se velika pažnja mora usredotočiti na njeno gospodarenje. Budući da neprestano rastu sve veći korisnički zahtjevi, a naročito zahtjevi u pogledu gospodarenja memorijom, nastojanja su bila kako maksimalno iskoristiti memoriju, ali sa što manje troškova te čim veću brzinu korištenja. Memorija postaje neophodna za normalno funkcioniranje računalnih sustava i čini njihov centralni najvažniji dio sustava.

U suvremenim računalnim sustavima moguće je imati velik broj pokrenutih programa, koji zauzimaju puno više memorijskog prostora nego što je to fizički raspoloživo. U mnoštvu sve složenijih aplikacija zahtijeva se sve više memorije u računalnim sustavima, a informatički stručnjaci htjeli bi imati na raspolaganju beskonačno veliku i brzu memoriju na svojim računalima. Zato u današnje vrijeme imamo podijeljenu memoriju prema hijerarhiji, a svaki operacijski sustav ima zadatak dobro koordinirati upotrebu ovih memorija. Operacijski sustav mora izvršiti svoje zadatke u pogledu olakšanja uporabe računala i same organizacije djelotvornog iskorištavanja svih dijelova računala. Suvremeni računalni sustavi upotrebljavaju procesore s takvim spremničkim međusklopovima, koji omogućuju da operacijski sustavi djelotvorno gospodare spremničkim memorijskim prostorom.

Razvojem suvremenih računalnih sustava u računalima se neprestano povećavao kapacitet glavne memorije te obujam mnogih programa u usporedbi sa dostupnom instaliranom memorijom. Ovaj problem obujma kapaciteta memorije u računalnim sustavima informatičari su pokušali riješiti na primjeni tehnike preklapanja, gdje se rješenje vidjelo u dijeljenju programa na nekoliko dijelova. Tako je u memoriji uvijek prisutan jedan od ovih dijelova, i on upravlja punjenjem drugih dijelova sekundarne memorije (disk, traka, itd.) u glavnu memoriju. Na tržištu informatičke opreme prisutna je visoka cijena brze glavne memorije, koja nameće potrebu da se glavna memorija proširi sve više sa jeftinom i sporom sekundarnom memorijom, koja koristi za zapis diskove, trake i dr. S aspekta gospodarenja memorijskim prostorom, informatički stručnjaci danas se sve više susreću sa problemima dijeljenja memorije i usklađivanja brzina rada procesora i glavne memorije. Neprestano proširenje memorijskog kapaciteta i što veće ubrzanje pohrane podataka traži da se računalni sustavi današnjice

neprestano razvijaju i nadopunjavaju. Vrhunac razvoja informacijsko – komunikacijskih tehnologija stvorio je takve računalne sustave, koji omogućuju da operacijski sustavi djelotvorno gospodare spremničkim memorijskim prostorom i dobro koordiniraju upotrebu memorija prema hijerarhiji.

Može se zaključiti da gospodarenje memorijskim prostorom nije jednostavan zadatak operacijskog sustava, budući da računalni sustavi mogu izvoditi programe čiji su procesni adresni prostori veći od fizičkog adresnog prostora računala.

## LITERATURA

1. Budin, L., Golub, M., Jakobović, D., Jelenković, L., *Operacijski sustavi*, Element d.o.o., Zagreb, 2010.
2. Kiš, M., *Informatički rječnik za školu i dom*, Andromeda d.o.o., Rijeka, 2006.
3. Peruško, U., *Digitalna elektronika logičko i električko projektiranje*, Školska knjiga, Zagreb, 1996.
4. Ribarić, S., *Arhitektura računala RISC i CISC*, Školska knjiga, Zagreb, 1996.
5. Turk, S., *Arhitektura i organizacija digitalnih računala*, Školska knjiga, Zagreb, 1990.
6. [www.ss-strukovna-djurdjevac.skole.hr/dokument?dm\\_document...](http://www.ss-strukovna-djurdjevac.skole.hr/dokument?dm_document...)

## POPIS SLIKA

|   |    |
|---|----|
| Slika 1. Memorijska hijerarhija.....  | 9  |
| Slika 2. Putovi podataka između glavne memorije, priručne memorije i procesora..... | 12 |
| Slika 3. Vanjska fragmentacija memorije.....  | 24 |
| Slika 4. Primjer operacije zbivanja.....  | 25 |



## SAŽETAK

Značaj uloge gospodarenja memorijskim prostorom postaje neopisivo važno u današnjem brzom razvoju cijelog računalnog sustava. Živimo u tzv. informatičkom dobu gdje se napretkom informatičke tehnologije proces upravljanja memorijom drastično poboljšao hardverski i softverski. U današnje vrijeme brz razvoj informacijsko – komunikacijskih tehnologija dostigao je vrhunac razvoja i uzdrmao sva područja ljudskog života, tako da paralelno rastu i sve veći korisnički zahtjevi, a naročito zahtjevi u pogledu gospodarenja memorijskim prostorom. Tokom povijesti javljali su se određeni problemi i nedostaci u rukovođenju memorijskim prostorom, i to je pokazalo da se velika pažnja mora usmjeriti na njeno gospodarenje. Nastojanja su neprestano bila kako maksimalno iskoristiti memoriju, ali sa što manje financijskih troškova i što veću brzinu korištenja. U suvremenim računalnim sustavima moguće je imati velik broj pokrenutih programa, koji zauzimaju puno više memorijskog prostora nego što je to fizički raspoloživo. Zbog toga informatičari na svojim računalima željeli bi imati na raspolaganju beskonačno veliku i brzu memoriju.

Memorija čini centralni dio suvremenih računalnih sustava i neophodna je za njegovo normalno funkcioniranje. Može se definirati kao sredstvo za spremanje podataka i kao funkcijska jedinica digitalnog računalnog sustava ona prima, čuva i predaje digitalne podatke. Memorija kao jedna od osnovnih dijelova računala, jedno je veliko polje okteta ili riječi, koji imaju svaki svoju adresu, gdje procesor dohvaća naredbe iz memorije. Rezultati tih naredbi mogu biti dodatni pristupi memoriji kako bi se dohvatilo operande, odnosno dio naredbe koji određuju radnju što će se obaviti nad određenim podatkom – operandom i pohranilo rezultate operacija.

Brz razvoj informatičke tehnologije doveo je do toga da suvremeni računalni sustavi upotrebljavaju procesore sa takvim spremničkim međusklopovima, koji omogućuju da operacijski sustavi djelotvorno gospodare spremničkim memorijskim prostorom. Računalni sustavi imaju podijeljenu memoriju prema hijerarhiji, a svaki operacijski sustav mora izvršavati svoje zadatke, koje se ogleda u olakšanju uporabe računala i organizaciji djelotvornog iskorištavanja svih dijelova računala te dobrom koordiniranju upotrebe ovih memorija. U računalnim sustavima memorijsko – upravljačka jedinica je zadužena za upravljanje memorijom. Kako bi se osigurao nesmetan rad cijelog računalnog sustava, neophodno je potrebno što efikasnije dodijeliti različite dijelove memorije što postaje glavni problem pri upravljanju memorijom. Korisnički programi i mnoštvo raznovrsnih aplikacija postaju sve složeniji i zahtjevniji te sve više nedostaje količine memorije koja je je dodijeljena određenom programu.

U glavne funkcije sustava za gospodarenje memorijom spada: vođenje evidencije o raspoloživosti slobodnih dijelova memorije, alokacija tj. dodjela memorije procesima, oslobađanje memorije kada nije više potrebna procesima, zamjena sadržaja između glavne memorije i diska kada glavna memorija nije dovoljno velika za dani proces itd. Informatički stručnjaci kod djelotvornog gospodarenja memorijskim prostorom susreću se sa problematikom dijeljenja memorije i usklađivanja brzina rada procesora i glavne memorije te kako ostvariti proširenje memorijskog prostora. Osnovno pitanje budućeg informatičkog napretka postaje kako povećati količinu memorije u računalima, te se traže rješenja u proširenju memorijskog kapaciteta i što veće ubrzanje pohrane podataka.

Ključne riječi: gospodarenje memorijom, memorija, memorijski prostor, operacijski sustav, računalni sustav, podjela memorije.

## SUMMARY

The importance of the role of managing memory space becomes incredibly important in today's rapid development of the entire computer system. We live in so-called information age where due to the progress of information technology process of memory management drastically improved both in hardware and software. Today, the rapid development of information and communication technology reached its peak of development and shook all areas of human life, so that the parallel growth increases user demands, especially the requirements for the management of storage space. Throughout history, we were approached by a number of problems and shortcomings in the management of storage space, and it has shown that great care must be given to its management. Efforts are constantly been made to maximize memory use, but with as little financial cost and greater speed of use. In modern computer systems it is possible to have a large number of running programs which take up a lot more memory than is physically available. That is why computer scientists would like to have a large and infinitely fast memory at their disposal on computers.

Memory makes the central part of modern computer systems and is required for its normal functioning. It can be defined as a mean for data storage and a functional unit of a digital computer system that receives, stores and transmits digital data. The memory as one of the basic parts of the computer, is one large field or words, which each have its own address and where the processor fetches commands from memory. The results of these commands can be extra memory access in order to fetch operands, or part of commands that determine the action that will take place over certain data (operand) and store the results of operations.

The rapid development of information technology has led to the modern computer systems that use processors with such a tank interfaces which allow operating systems to efficiently manage storage space (tanks). Computer systems have a shared memory regarding the hierarchy, but each operating system must perform its tasks, and this is reflected in facilitating the use of computers and the organization of an effective utilization of computer parts and well-coordinated use of the memory. In computer systems memory – control unit is responsible for memory management. In order to ensure the smooth operation of the entire computer system, it is necessary to allocate different parts of the memory (which becomes a major problem in managing the memory) more efficiently. User programs and a multitude of diverse applications become more complex and demanding, and the amount of memory that is assigned to a specific program is increasingly lacking.

The main functions of the system for managing memory include: keeping records of the availability of parts of memory allocation, ie, memory allocation processes, freeing memory when processes are no longer needed, replacement of content between main memory and disk when main memory is not large enough for a given process, etc. Talking about effective management of memory space IT experts are faced with the issue of sharing memory and matching the speed of the processor and the main memory together with expanding additional memory. The basic question of the future of the IT progress becomes how to increase the amount of memory in computers, and they seek solutions to extend the memory capacity and accelerate data storage.

Keywords: management of memory, memory, memory space, operating system, computer system, memory division.