

Operativni sustavi: upravljanje memorijom

Delinger, Antun

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:137:211605>

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

Download date / Datum preuzimanja: **2025-02-27**



Repository / Repozitorij:

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



Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
„Dr. Mijo Mirković“

Antun Delinger

Operativni sustavi: upravljanje memorijom

Završni rad

Pula, 2016.

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
„Dr. Mijo Mirković“

Antun Delinger

Operativni sustavi: upravljanje memorijom

Završni rad

JMBAG: 0303043755

Studijski smjer: Poslovna informatika

Predmet: Ekonomska informatika

Mentor: prof.dr.sc. Vanja Bevanda

Pula, 2016.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ 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.

Student:

U Puli, . . . 2015.

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, *Antun Delinger* dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom *Operativni sustavi: upravljanje memorijom* 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.

Student:

Sadržaj

1. Računalni sustav.....	7
2. Uvod u operativne sustave.....	9
2.1. Povijest razvitka	10
2.1.2. <i>Početne generacije operacijskih sustava</i>	10
2.2. Model jednostavnog računala	11
2.2.1. <i>Von Neumannov model računala</i>	11
2.2.2. <i>Sabirnička građa računala</i>	13
2.3. Računalni proces	14
3. Ulazno – izlazne operacije	17
4. Programi, procesi i dretve.....	18
5. Upravljanje memorijom	19
5.1. Magnetski diskovi	19
5.1.1. <i>Organizacija zapisivanja sadržaja na disku</i>	20
5.1.2. <i>Vremenska svojstva diskova</i>	21
5.2. Razvoj načina dodjeljivanja radnog spremnika.....	23
5.2.1. <i>Statičko raspoređivanje radnog spremnika</i>	23
5.2.2. <i>Dinamičko raspoređivanje radnog spremnika</i>	25
5.3. Dodjeljivanje spremnika straničenjem	29
5.4. Novija kretanja memorijskih modula	31
5.4.1. <i>SSD disk</i>	32
5.4.2. <i>Nand flash memorija</i>	33
6. Datotečni podsustav.....	36
6.1. Struktura datoteka	37
7. Višediskovni zalihosni spremnici.....	39
8. Sigurnost računalnih sustava.....	40
8.1. Vrste napada na sigurnost	40
8.2. Osnove kriptografije	41
8.2.1. <i>Kriptiranje</i>	41
9. Zaključak	43
10. Literatura	44
11. Popis slika.....	45
12. Sažetak.....	46
13. Summary.....	47

Uvod

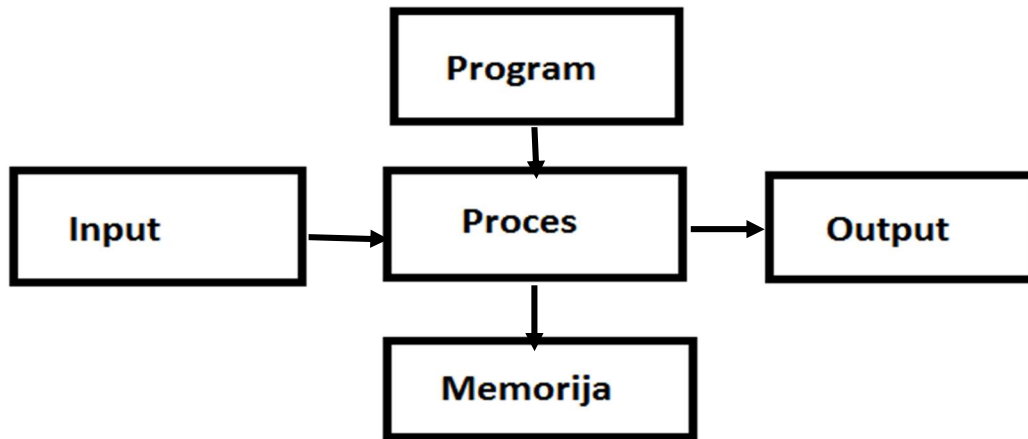
Računala su svojim dolaskom na tržište pokrenula revoluciju – rezultat je masovno uvođenje računala u sve grane ljudske djelatnosti, a osim u poslovne svrhe, računalo je našlo primjenu i u kućanstvima. Sve jednostavnija izrada i sve niža cijena računala dovela je do toga da je računalo nezaobilazno u svakom kućanstvu. Prihvatanjem računala kao dijelom svakodnevice, ljudi su dali zeleno svjetlo naglom razvoju IT industrije, a rezultat tog razvoja vidljiv je svakodnevno. Velika potražnja za računalima dovela je do razvoja raznih oblika računala – stolnih, prijenosnih, mobilnih i tako dalje. Osim što imaju vanjsko kućište i nekakav oblik grafičke reprezentacije svog sadržaja, računala imaju i svoju programsku komponentu, a najvažnija programska komponenta, odnosno program koji povezuje računalo i korisnika te omogućuje međusobnu interakciju naziva se operativni sustav. Operativni sustav predstavlja ključan faktor u radu računala te bez njega računala ne bi uspjela privući pažnju krajnjih potrošača budući da bi bila pretjerano komplicirana. Postoji više vrsta operativnih sustava za različita računala te zasebne inačice za zasebne zadatke koje će izvršavati (primjerice inačice za poslovne korisnike).

Ovaj rad donosi kratko upoznavanje s operativnim sustavima, načinima na koje rade te vrstama koje možemo susresti. Osim upoznavanja samih operativnih sustava, opisan je i princip rada jednostavnog računala. Operativni sustavi sami po sebi predstavljaju opsežno područje te je stoga ovaj rad orijentiran na nešto uže područje – upravljanje memorijom. Rad prikazuje način upravljanja memorijom, vrste memoriju koje se mogu naći u računalu te novija kretanja memorijskih spremnika.

Završni rad sastoji se od osam poglavlja tijekom kojih su razrađeni operativni sustavi s naglaskom na upravljanje memorijom. Prva dva poglavlja donose uvid u osnovni tip računalnog sustava te osnove rada operativnih sustava. Kasnijim poglavljima detaljnije se ulazi u tematiku operativnih sustava, a u petom, šestom te sedmom poglavlju obrađeno je upravljanje memorijom. Nadalje, ukratko je opisana važnost sigurnosnog aspekta računalnih sustava te od čega se ono sastoji, a na samom kraju dano je zaključno razmatranje autora završnog rada.

1. Računalni sustav

Računalni sustav je sustav sposoban preuzeti set inputa, procesirati ih te kreirati set outputa. To obavlja kombinacijom hardvera i softvera.



Slika 1. Osnovni oblik računalnog sustava

(Izvor: izradio student)

Računalni sustav posjeduje jedan ili više inputa kako bi osigurao podatke. TI podaci su zatim procesirani na neki način. Rezultat procesiranja poslan je na izlazni uređaj (output) ili je privremeno uskladišten prije slanja na izlazni uređaj. Kako bi se transformacijski proces (input-output) mogao izvršiti, potreban je set instrukcija koje govore što treba biti napravljeno. Taj set instrukcija naziva se program.¹

U transformacijskom procesu, input predstavlja podatke ili upute koji u transformacijski proces ulaze putem ulaznih uređaja (tipkovnica, miš, itd.) Zadatak inputa je osigurati podatke za daljnju obradu. Proces predstavlja fazu u kojoj se primljeni podaci transformiraju u smislene informacije. Rezultat ove faze je output. Output predstavlja fazu gdje su informacije predstavljene korisniku u prikladnog formatu. Većina outputa podrazumijeva pretvorbu digitalnih podataka u grafički ili zvučni prikaz što ih čini vidljivima na uređajima kao što su računalni zaslone, zvučnici, pisač itd.²

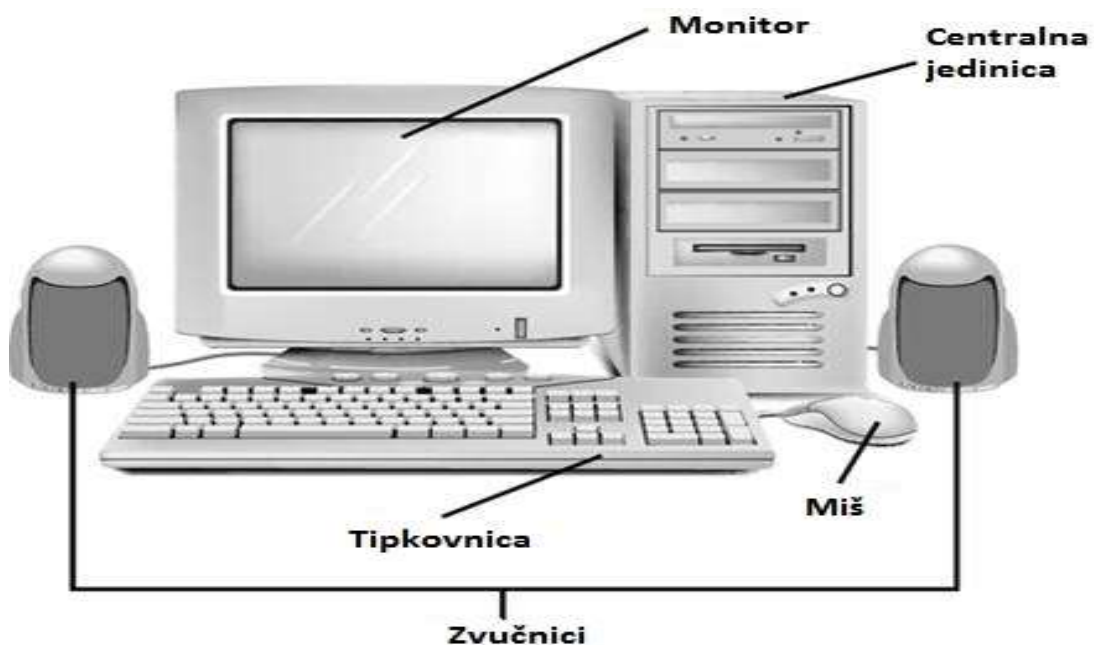
¹ http://www.teach-ict.com/gcse_computing/ocr/211_hardware_software/computer_system/miniweb/pg2.htm, 15. kolovoz 2016

² <http://passnownow.com/classwork-exercise-and-series-computer-jss-2-the-computer-system/> 15. kolovoz 2016

Računalni sustav sastoji se od centralne jedinice računala (Computer System Unit), ulaznih uređaja (miš, tipkovnica) te izlaznih uređaja (zaslon računala, zvučnici, pisač itd.) Centralna jedinica računala predstavlja najvažniji dio računalnog sustava te se pomoću nje povezuju svi ostali elementi. Unutar računalnog sustava, centralna jedinica predstavlja alat za transformaciju inputa u outpute.

Centralna jedinica sastoji se od 6 osnovnih elemenata:

- CPU (Central Processing Unit) – procesor – glavni dio računala vođen zadanim programskim naredbama izvodi osnovne radnje nad podacima.
- GPU (Graphics Processing Unit) – grafička kartica – procesor specijaliziran za prikazivanje računalne grafike
- MBO (Motherboard) – matična ploča – povezuje dijelove računala u jednu cjelinu
- PSU (Power Supply Unit) – napajanje računala- osigurava napon i struju računala
- HDD (Hard Disk Drive) – tvrdi disk računala – trajna pohrana podatak u računalu
- RAM (Random Access Memory) – radna memorija – privremena pohrana podataka



Slika 2. Primjer računalnog sustava

(Izvor: <http://flylib.com/books/en/2.944.1.11/1/>, 15. kolovoz 2016)

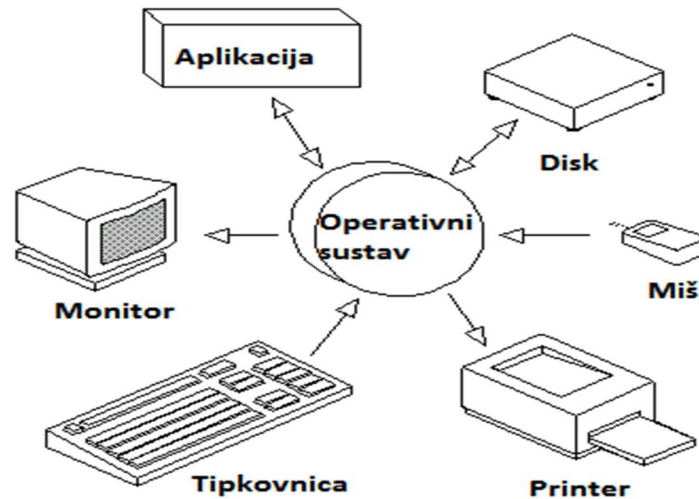
Slika 2. prikazuje računalni sustav u osnovnom obliku – ulazni uređaji pomoći kojih korisnik daje naredbe, centralna jedinica (System Unit) pomoću koje se obrađuju naredbe te izlazni uređaji pomoću kojih se prikazuje krajnji rezultat procesa. Osim vidljivih elemenata računalnog sustava, postoji još i softverska komponenta. Skup osnovnih softvera naziva se operativni sustav čiji je zadatak upravljanje radom računalnog sklopovlja, odnosno operativni sustav olakšava komunikaciju između korisnika i računala.

2. Uvod u operativne sustave

Operativni sustav najvažniji je program koji se nalazi na računalu. Svako računalo mora imati operativni sustav kako bi moglo pokretati programe i aplikacije.

Računalni operativni sustavi izvode osnovne zadatke, kao što je prepoznavanje inputa (ulaznih signala) sa tipkovnice ili slanja outputa (izlaznih signala) na jedinicu prikaza (monitor), prateći podatke i direktorije na disku te kontrolirajući periferne uređaje kao što je pisač. Za veće sustave, operativni sustav ima još veću moć i odgovornost. Ponaša se kao prometni policajac – osigurava da korisnici i programi koji rade istovremeno ne sputavaju jedni druge. Operativni sustav je također zadužen i za sigurnost – osigurava da neautorizirani korisnici ne mogu pristupiti sustavu.³

³ http://www.webopedia.com/TERM/O/operating_system.html, 15. kolovoz 2016



Slika 3. Operativni sustav (OS) dijagram

(Izvor: http://www.webopedia.com/TERM/O/operating_system.html, 19. kolovoz 2016.)

2.1. Povijest razvitka

Na samom početku pojave prvih računala u kasnim 1930-tim i tijekom Drugog svjetskog rata nisu postojali operacijski sustavi te su računala bila programirana na razini stroja. Kod nulte generacije računala bilo je moguće izvršavati samo jedan program koji je imao potpunu kontrolu nad računalom. Sve rutine za obavljanje nižih operacija (kao čitanje s papirne vrpce ili ispisivanje na pisaču) bile su sastavni dio svakog programa. Za programiranje nulte generacije računala programer je morao dobro poznavati sklopovlje računala, a vještina programiranja u sebi je sadržavala određenu količinu prepravki na sklopovlju: premještanje žica, dodavanje ili oduzimanje elektroničkih sklopova. Nulta generacija računala bili su eksperimentalni, novi uređaji, i kao takvi prepravke na sklopovlju bile su nužne jer nije postojala dovoljna količina znanja i iskustva kako graditi i programirati računala kao i o mogućim praktičnim primjenama računala i računarstva.⁴

2.1.2. Početne generacije operacijskih sustava

Prvi operacijski sustavi razvijeni su tijekom razvoja prve generacije računala u 50-tim godinama 20. stoljeća, a ovaj razvoj bio je moguć zbog povećanih sposobnosti prve

⁴ <https://loomen.carnet.hr/mod/book/view.php?id=181686&chapterid=41751>, 19. kolovoz 2016.

generacije te zbog sakupljenog iskustva u proizvodnji i programiranju računala tijekom izrade nulte generacije računala.

U prvoj generaciji koju se vremenski smješta od 1945. do 1955., nije bilo operacijskih sustava. Sklopovski su ondašnja računala karakterizirale vakuumske cijevi.

U drugoj generaciji pojavile su se operacijski sustavi kad se izvodila batch obrada. Vremenski se tu generaciju smješta od 1956. do 1965. godine. Sklopovski ju karakteriziraju tranzistori.

Treću generaciju se vremenski smješta od 1965. do 1980. godine. Sklopovski ju karakteriziraju (integrirani krugovi). Svojstveno joj je multiprogramiranje i ostalo. Iz tog vremena datiraju MULTICS, UNIX, Minix i Linux.

Četvrta generacija teče od 1980. do danas. Sklopovski ih karakteriziraju sve integrirani čipovi. Od tada datiraju CP/M, DOS, MS-DOS, sustavi Windows i ostali.⁵

2.2. Model jednostavnog računala

2.2.1. Von Neumannov model računala

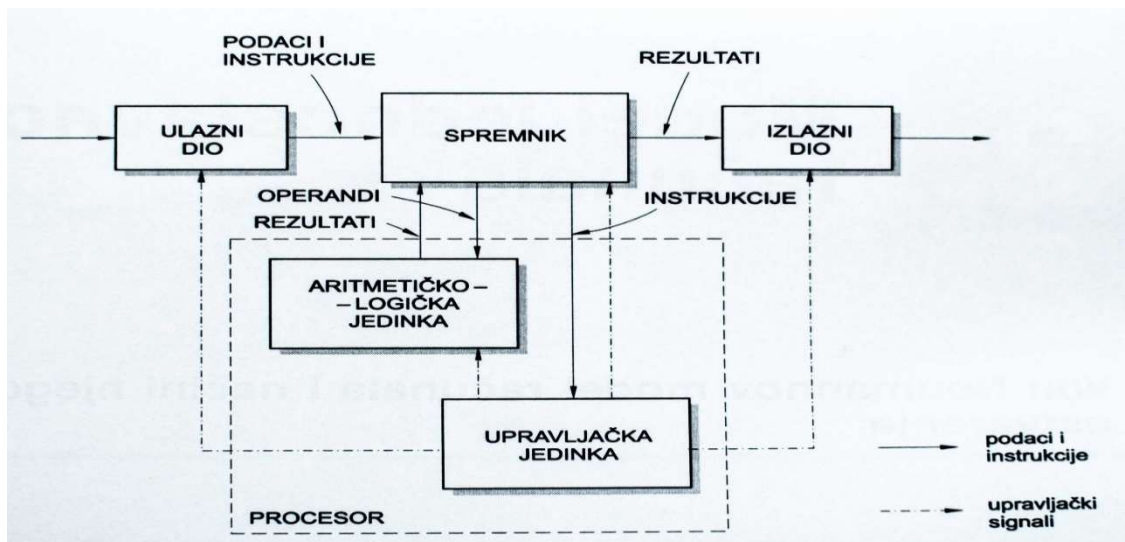
Današnji računalni sustavi zasnivaju se još uvijek pretežito na koncepcijskom modelu koji je još 1945. godine opisao John Von Neumann.

Von Neumannov model utvrđuje da svako računalo mora imati sljedeće dijelove:

- ulazni dio preko kojeg se u spremnik unose iz okoline podaci i instrukcije programa
- izlazni dio preko kojeg se u okolini prenose rezultati programa
- radni ili glavni spremnik u koji se pohranjuju svi podaci i instrukcija programa uneseni izvana, kao i rezultati djelovanja instrukcija

⁵ loc.cit.

- aritmetičko-logičku jedinku koja može izvoditi instrukcijama zadane aritmetičke i logičke operacije
- upravljačka jedinka koja dohvaća instrukcije iz spremnika, dekodira ih i na temelju toga upravlja aritmetičko – logičkom jedinkom, te ulaznim i izlaznim dijelovima



Slika 4. Funkcijski Von Neumannov model računala

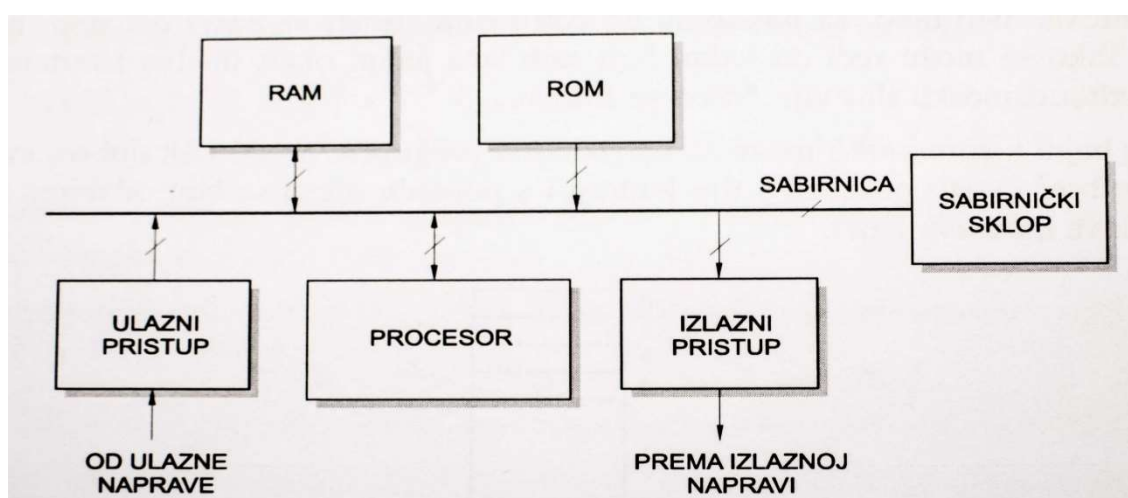
(Izvor: 1. Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., str9.)

Slika 4. ilustrira međusobnu povezanost svih tih dijelova. Na slici su označeni tokovi podataka, instrukcija i upravljačkih signala.

Središnji dio računala je spremnik. U njega se slijevaju svi podaci i instrukcije koje se unose u računalo preko ulaznog dijela, te svi rezultati operacija iz aritmetičko-logičke jedinice. Preko izlaznog dijela rezultati izračunavanja prenose se u okolinu. Iz spremnika upravljačka jedinka dohvaća instrukcije i na temelju njih upravlja preostalim dijelovima računala. Upravljačka jedinka određuje koju će operaciju izvesti aritmetičko-logička jedinka. Upravljačka jedinka i aritmetičko-logička jedinka spregnute su današnjim računalima u jednu cjelinu i, dodatno, s jednim skupom registara čine procesor.

2.2.2. Sabirnička građa računala

U funkcijskom prikazu računala na slici 4. vidljivo je kako su pojedini dijelovi računala međusobno povezani. Svaka od crta koja predstavlja tok podataka, instrukcija ili upravljačkih signala sastoji se od većeg broja vodiča preko kojih se prenose električki signali kojima se prenose bitovi. Takvo isprepletano međusobno povezivanje pojedinih dijelova računala nespretno je i stoga je osmišljen sabirnički sustav za njihovo povezivanje. Sabirnica je jedan zajednički snop vodiča na koji su spojeni svi dijelovi računala. Osim vodiča sabirnica ima i svoj sabirnički elektronički sklop, koji pomaže pri ostvarivanju veza.



Slika 5. Sklopovlje računala povezano sabirnicom

(Izvor: . Budin L., G., op.cit., str. 11.)

Računalo sa sabirničkim povezivanjem dijelova prikazano je na slici 5. Sve potrebne razmjene podataka, instrukcija i upravljačkih signala obavljaju se preko zajedničkih vodiča. Kosa crtica preko crte koja predstavlja sabirnicu označava da je riječ o snopu vodiča. Jasno je da se preko zajedničkih vodiča ne može uspostavljati više istovremenih veza već se mogu obavljati samo pojedinačni prijenosi. Sabirnica se stoga mora naizmjenice – s podjelom vremena (engl. Time share) – koristiti za ostvarenje potrebnih veza između dijelova računala.⁶

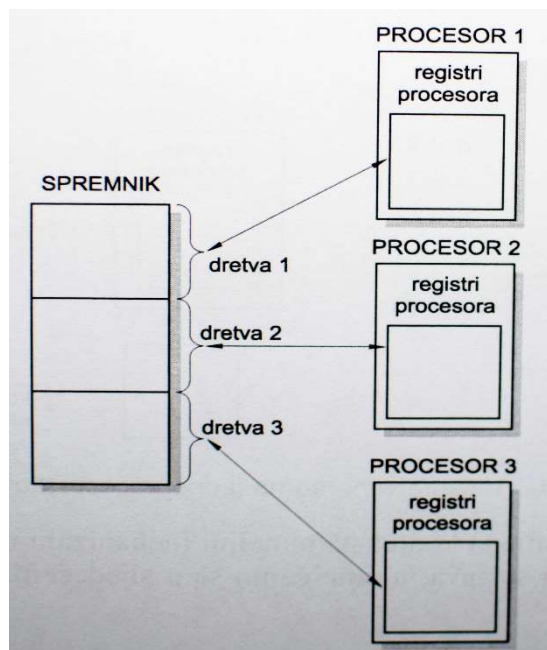
⁶ . Budin L., G., op.cit., str.11.)

2.3. Računalni proces

Program koji je zapisan na papiru ili pohranjen u strojnom obliku u spremniku računala statička je tvorevina – niz instrukcija koji izvoditelj programa, tj. Procesor mora razumjeti i znati obaviti. Međutim, kada se program počinje izvoditi, njemu se mogu pripisati neka svojstva:

- trenutak početka izvođenja programa
- trenutak završetka izvođenja programa
- trajanje izvođenja programa
- zaustavljanje izvođenja programa i sl.

Time izvođenje programa dobiva obilježje procesa. S obzirom na to da se proces odvija u računalu govorimo o računalnom procesu ili, ako je jasno da je riječ o računalnom procesu, onda možemo kraće reći samo proces. Prije započinjanja nekog procesa u računalu se moraju stvoriti uvjeti za njegovo izvođenje. Osnovni je zadatak operacijskog sustava stvaranje takvih uvjeta za odvijanje programa u kojima se izvođenje može opisati ovakvim modelom koji se zasniva na opisu rudimentarnog računala. Prema tome, u stvarnim računalnim sustavima možemo ponašanje programa promatrati kao proces. Unutar procesa mora postojati barem jedna instrukcijska dretva. Operacijski sustav mora uspostaviti sve uvjete za odvijanje procesa. Između ostalog, on mora osigurati nesmetano izvođenje dretve, tj. Osigurati spremnički prostor i procesor u skladu s modelom našeg rudimentarnog računala. Brzina izvođenja dretve određena je brzinom procesora. Katkada će se izvođenje dretve morati na tren zaustaviti i kasnije nastaviti.



Slika 6. višedretveni rad na višeprocorskom računalu

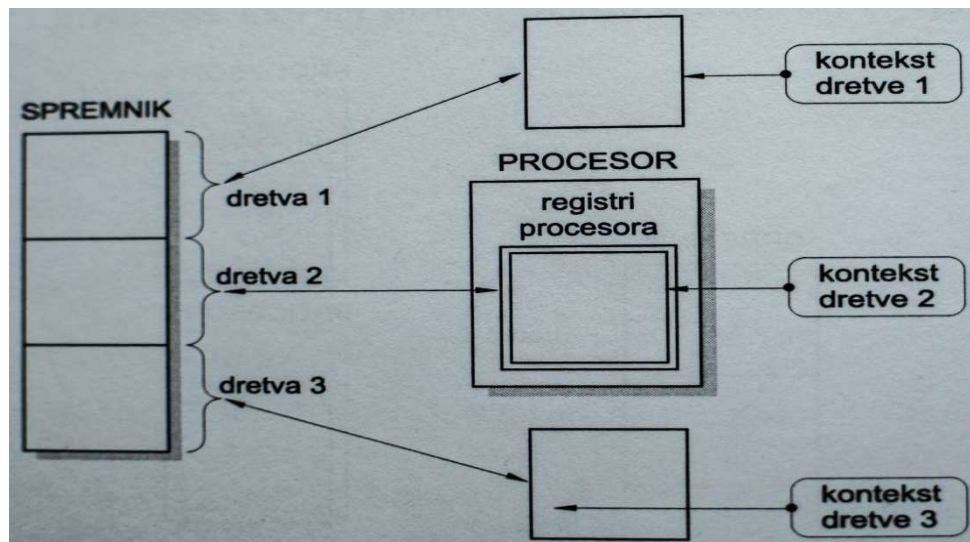
(Izvor: Budin L., G., op.cit., str.31)

U računalima s više procesora (u višeprocorskim računalima) proces bismo mogli podijeliti na više dretvi. Spremnik bi tada trebalo podijeliti tako da svaka dretva dobije svoj dio adresnog prostora iz kojeg pripadni procesor dohvaća instrukcije i podatke. Svaki procesor „provlačio“ bi svoju dretvu kroz instrukcije i tako neki zajednički posao može biti brže obavljen (slika 6.). Kažemo da takva računala podržavaju višedretveni rad (engl. multithreading).

U današnje vrijeme gotovo svi (višenamjenski) računalni sustavi podržavaju višedretveni rad, što zbog jednostavnosti izvedbe primjenskih programa, što zbog povećanja učinkovitosti iskorištavanja svih dostupnih resursa sustava (npr. iskorištavanje svih procesorskih jedinki višestrukog procesora na čipu).

Štoviše, višedretveni se rad može provesti i u jednoprocorskom računalu, i to tako da taj jedan jedini procesor naizmjenice „provlači“ jednu od više dretvi. U tom se slučaju ne može govoriti ubrzavanju odvijanja procesa, ali se može postići da procesor izvodi jednu od dretvi za vrijeme dok druga mora zbog nekog razloga čekati. S obzirom na to da svaka dretva za izvođenje treba „svoj procesor“, u jednoprocorskom sustavu mora se, pri prebacivanju izvođenja s jedne dretve na drugu, osigurati da svaka dretva radi sa svojim skupom registara. To se može postići tako da se sadržaj registara procesora one dretve čije se izvođenje želi prekinuti pohrani na neko rezervirano

mjesto u spremniku, a u registre procesora smjesti sadržaje koji pripadaju dretvi čije izvođenje upravo treba započeti. Ako se takva promjena sadržaja registara obavi ispravno, onda se stvara privid da svaka dretva posjeduje „vlastiti“ procesor. Sadržaj registara procesora zovemo kontekstom dretve, a promjenu sadržaja registara promjenom konteksta.⁷



Slika 7. Višedretveni rad na jednoprocesorskom računalu

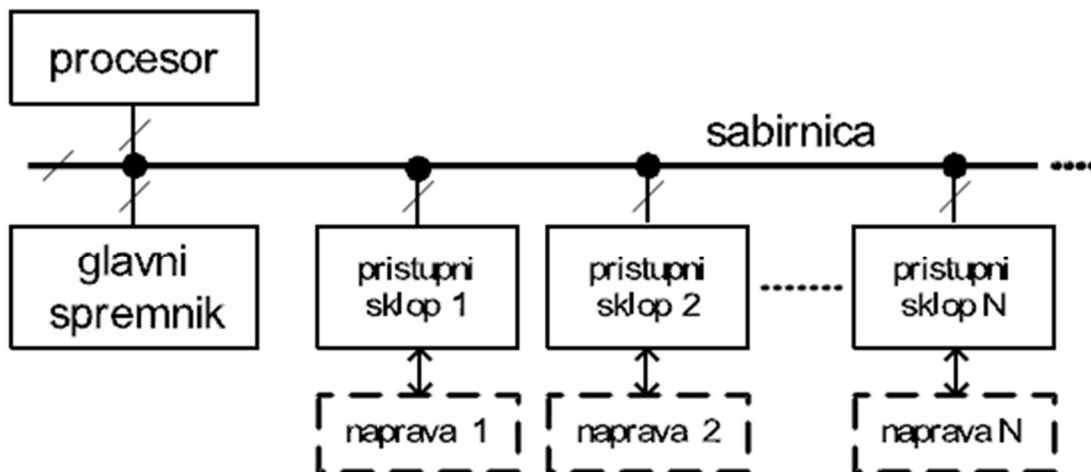
(izvor: Izvor: Budin L., G., op.cit., str32.)

⁷ Ibidem str. 32

3. Ulazno – izlazne operacije

Uobičajene ulazne naprave su tipkovnica i miš, dok najčešće izlazne naprave predstavljaju monitor i pisač. Sve ulazne i izlazne naprave nazivamo zbirnim imenom ulazno-izlazne naprave (UI-naprave) ili periferijske naprave. U skupinu tih naprava možemo ubrojiti i vanjske spremnike, koji su i ulazne i izlazne naprave, a tu spadaju: magnetski diskovi, magnetske diskete te optički diskovi. Osim ovih osnovnih naprava današnja računala mogu se opremiti raznovrsnim ulazno-izlaznim napravama i uređajima kao što su: naprave za čitanje znakova i slika, naprave za crtanje i izradu slika, videokamera, videorekorder, kompaktni disk (CD), mikrofon, zvučnici itd. ⁸

Navedene ulazno-izlazne naprave koje se spajaju u računalo razlikuju se u načinu i brzini rada te se iz tog razloga naprave ne spajaju izravno na glavnu sabirnicu. Naprava je spojena na međusklop, odnosno pristupni sklop računala koje je s jedne strane prilagođen toj napravi, a s druge strane prilagođava se protokolima sabirničkog sustava. Također, pristupni sklop omogućava sinkronizaciju rada procesora i ulazno-izlaznih naprava. Spajanje ulazno-izlaznih naprava na sabirnicu računala ilustrirano je na slici 8.



Slika 8. Spajanje ulazno-izlaznih naprava na sabirnicu računala (Izvor: Jelenković: Operacijski sustavi – skripta 2014./2015.)

⁸ Ibidem, str. 33

4. Programi, procesi i dretve

Računalo obavlja neki korisni zadatak (engl. task) tako da izvodi programe pripremljene u višem programskom jeziku. Programi određuju sve aktivnosti koje računalni sustav treba obaviti, način unošenja ulaznih podataka, način prikaza rezultata, te način njihova trajnijeg pohranjivanja.

Kada se program preveden u strojni oblik pokrene, on dobiva i neke vremenske attribute, kao što su: trenutak početka, trajanje izvođenja, trenutak završetka. Tijekom svog izvođenja program može zastati i biti ponovno pokrenut. Prema tome, program koji se izvodi djeluje tako da na uredni način utječe na promjene stanja računalnog sustava.

S obzirom na to da pojedini procesi u raznim fazama svojeg odvijanja raznoliko troše pojedine dijelove računalnog sustava, pokazalo se razumnim posao organizirati tako da se u istom vremenskom razdoblju izvodi više zadataka. Dakle, više procesa može istodobno napredovati ako se odvijaju u različitim dijelovima računalnog sustava. Primjerice, ako jedan proces čeka na završetak ulazne operacije kako bi mogao nastaviti neko izračunavanje, za to vrijeme drugi proces može izvoditi svoje instrukcije. Pojedine, jasno razlučive dijelove računala možemo nazvati računalnim sredstvima ili samo sredstvima (engl. resource). Kada se govori o sredstvima, onda to nisu samo pojedini sklopovski dijelovi već i neki programski dijelovi kao i neki podaci. Proces se mogu odvijati i tako se neko sredstvo naizmjenice dodjeljuje pojedinom od procesa, pa oni samo prividno istodobno napreduju. Takav višezadačni ili višeprogramski rad (engl. multitasking) omogućuje s jedne strane bolje iskorištenje svih sredstava računalnog sustava, a s druge strane olakšava i organizaciju poslova koji se obavljaju računalnim sustavom. Odvijanje procesa obavlja se izvođenjem njegova niza instrukcija, tj. njegove dretve. U svakom procesu mora, prema tome, biti prepoznatljiva barem jedna dretva. Proces se obavlja tako da procesor izvodi tu dretvu. Pokazalo se, međutim, praktičnim pojedine zadatke dijeliti na podzadatke kako bi se s jedne strane lakše svladala složenost zasnivanja i izgradnje programskih sustava, a s druge strane omogućilo bolje iskorištenje računalnih sredstava sustava i to unutar jednog procesa. Stoga su u suvremen operativne sustave uvedeni mehanizmi koji podržavaju izvođenje procesa s više dretvi pa govorimo o višedretvenom načinu rada ili višedretvenosti (multithreading). Dakle suvremeni su operativni sustavi višezadačni i višedretveni.

Dretve koje djeluju unutar jednog procesa dijele sva sredstva koja je operativni sustav stavio na raspolaganje tom procesu. Posebice se to odnosi na adresni prostor procesa. Prema tome, sve dretve istog procesa mogu neposredno pristupiti do svih adresa adresnog prostora svog procesa. Međutim, dva procesa ne mogu jedan drugom neposredno adresirati varijable ako je potrebno obaviti razmjenu podataka između dva procesa onda se to mora čini posredno s pomoću mehanizama koje osigurava operativni sustav.⁹

5. Upravljanje memorijom

Upravljanje memorijom funkcija je operativnog sustava koja upravlja primarnom memorijom i premješta procese naprijed – natrag između glavne memorije i diska između izvođenja.

Upravljanje memorijom prati svaku memorijsku lokaciju, neovisno o tome je li ona dodijeljena nekom procesu ili je slobodna. Upravljanje memorijom odlučuje koji proces će dobiti memoriju i u koje vrijeme te prati memoriju ne bi li ostala slobodna ili nedodijeljena te sukladno tome joj ažurira status.¹⁰

5.1. Magnetski diskovi

Magnetski diskovi proizvode se u sljedeće dvije izvedbe:

- jednoj koju na hrvatskom zovemo diskom (govori se u skladu s engleskim i tvrdi disk)
- drugoj koju na hrvatskom zovemo disketom (govori se u skladu s engleskim i meki disk) (*engl. hard disk*)

Diskovi se na računalo priključuju s pomoću prikladnog upravljačkog sklopovlja. Jedinka magnetskog diska (*engl. magnetic disk drive*) sastoji se od dvije komponente:

⁹ Ibidem, str. 61.

¹⁰ http://www.tutorialspoint.com/operating_system/os_memory_management.htm, 22 kolovoz 2016.

- elektromehaničkog dijela koji čine jedna okrugla ploča ili više njih presvučenih magnetskim materijalom koje se vrte konstantnom brzinom i mehanizma magnetskih glava koje se mogu pomicati približno radijalno iznad tih ploča
- upravljačkog sklopa sastavljenog od mikroprocesora (mikroupravljača), spremnika, sučelja prema elektromehaničkom dijelu i sučelja prema sabirnici računala koje se ponaša kao pristupni sklop s neposrednim pristupom radnom spremniku.

5.1.1. Organizacija zapisivanja sadržaja na disku

Na slici 9. prikazan je disk s više ploča. Sve ploče učvršćene su na istu osovinu i vrte se konstantnom brzinom ω . U smjeru radiju ploča pomiču se glave za pisanje i čitanje. Glave se mogu vrlo precizno postaviti na jedno mjesto i tada ispod njih prolazi vrlo uski prsten ploče koji zovemo stazom (engl. track). Staze jednakih polumjera svih diskova leže na zamišljenom plaštu valjka i zovemo ih zajedničkim imenom cilindar. Svaka staza podijeljena je na jednake dijelove (kružne lukove s jednakim središnjim kutovima koje zovemo sektorima. S pomoću magnetskih glava promjenom se magnetskog toka upisuju bitovi u magnetski materijal površine ploče. Taj se sadržaj kasnije može pročitati jer magnetizirana površina diska inducira električke signale u glavi za čitanje iz kojih se razaznaje upisani sadržaj.

U jedan sektor može se pohraniti određeni broj bitova. Uobičajeno je 512, 1024 ili 4096 bajtova. Svaki sektor na disku ima svoju jedinstvenu adresu koja se izračunava iz:

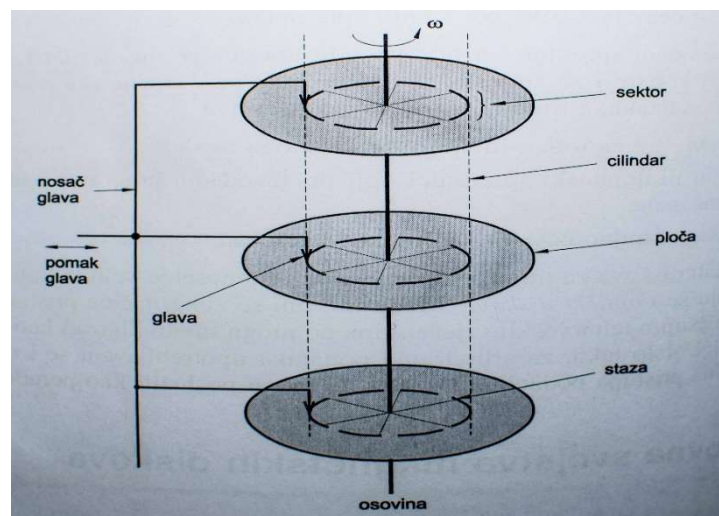
- rednog broja ploče
- rednog broja staze na ploči
- rednog broja sektora na stazi

Ploča se odabire aktiviranjem odgovarajuće glave, redni broj staze određuje se na temelju položaja glava, a redni broj sektora dobiva se tako da se mjere kutovi vrtnje počevši od posebne oznake na ploči koja označava početni kut. Na disku se ne adresiraju pojedinačni bajtovi nego blokovi bajtova. Blokovi bajtova sastoje se od jednog ili više sektora. U uobičajenim izvedbama diskova svaka ploča ima svoju glavu za pisanje, odnosno čitanje, ali se u jednom času može pristupiti samo do jedne ploče, tj. može se čitati ili pisati samo u jednu stazu cilindra.

Upravljački program za upravljanje diskom prenosi iz glavnog spremnika na disk, ili obrnuto, blokove bajtova koji su jednake veličine kao i sektori diska ili im je veličina cjelobrojni višekratnik veličine sektora. Pritom se programski mora odrediti početna adresa bloka u radnom spremniku s jedne strane te redni broj sektora na disku s druge strane. Nakon toga može početi prijenos kojega pristupni sklop obavlja neposrednim pristupom do spremnika.

Svojstva diskova:

- promjeri ploča kreću se u granicama od 1 do 8 inča (najčešće 3.5 inča)
- gustoća staza iznosi od 3000 do 250 000 staza po jednom inču radijusa
- diskovna jedinica može imati od 1 do 20 ploča
- broj staza na disku kreće se od 1500 do 400 000
- kapaciteti diskova kreću se u granicama od nekoliko stotina megabajta do nekoliko terebajta
- ploče diska čvrsto su ugrađene u diskovne jedinice i ne mogu se zamijenjivati¹¹



Slika 9. Mehanički izgled diska (Izvor: Budin L., G., op.cit.,str. 188.)

5.1.2. Vremenska svojstva diskova

Vremenska svojstva diskova određena su pretežito brzinama pokretanja njegovih mehaničkih dijelova. Vrijeme potrebno za prijenos sadržaja nekog sektora s diska ili pohranjivanje sadržaja u taj sektor može se podijeliti na dva dijela i to na:

¹¹ Izvor: Budin L., G., op.cit., str. 187.

- trajanje postavljanja glave (engl. head positioning time)
- trajanje prijenosa podataka (engl. data transfer time)

Trajanje postavljanja glave može se nadalje podijeliti na dvije komponente i to:

- trajanje traženja staze, ili kraće: trajanje traženja (engl. seek time)
- rotacijsko kašnjenje (engl. rotational latency)

Trajanje traženja ovisi o razmaku između početnog položaja glave i staze na koju se glava mora postaviti. Ako se glava pomiče samo za nekoliko staza, trajanje se uglavnom svodi na fino pozicioniranje. Kod kratkih pomaka ručica se u prvom dijelu ubrzava i mora se početi usporavati prije nego li dosegne maksimalnu brzinu. Za veće pomake nakon početnog ubrzavanja dolazi u razdoblje konstantne brzine, nakon kojeg slijedi usporavanje i konačno fino pozicioniranje. Uobičajeno se za vrijeme traženja uzima vrijeme koje je potrebno za prijelaz preko trećine ukupnog broja staza.

Rotacijsko kašnjenje nastaje zbog toga što se nakon postavljanja glave na odabranu stazu mora prije početka prijenosa pričekati da se ispod glave za čitanje pojavi traženi sektor. U najlošijem slučaju može se dogoditi da je adresirani sektor upravo prolazio ispod glave kad je ona prispejal na stazu i tada se mora čekati puni okretaj. Najpovoljniji je slučaj onaj kada traženi sektor upravo nailazi ispod glave u trenutku njezina postavljanja na stazu. Za rotacijsko kašnjenje uzima se prosječna vrijednost tih dvaju krajnjih slučajeva.

Trajanje prijenosa podataka određeno je brzinom prijenosa (engl. data transfer rate) i količinom prenesenih podataka. Brzina prijenosa određena je brzinom kojom ispod glave za čitanje ili pisanje promiču bajtovi sektora. Brzina prijenosa može se, dakle, izračunati tako da se broj sektora na stazi pomnoži s brojem bajtova u sektoru i taj umnožak podijeli s trajanjem jednog okretaja. Ovdje treba naglasiti da je unutar jednog sektora broj zapisanih bitova mnogo veći od broja pohranjenih bajtova jer se uz informacijske bitove na disk pohranjuju i dodatni bitovi zaštitnog kodiranja. Pri određivanju brzine prijenosa, promatramo samo korisne informacijske bitove, odnosno bajtove.

Napredak u tehnologiji proizvodnje diskova omogućio je da se tijekom zadnjih desetak godina zamjetno povećao kapacitet diskova, ali su poboljšanja vremenskih

svojstava mnogo skromnija. Prosječno trajanje postavljanja glava opalo je s vrijednosti od približno 20ms na ispod 10ms. Isto tako, tipična brzina okretaja povećana je s 3600 okretaja u minuti na 5400 do 15 000 okretaja u minuti. Brzine prijenosa povećale su se do oko 100 mb/s.¹²

5.2. Razvoj načina dodjeljivanja radnog spremnika

5.2.1. Statičko raspoređivanje radnog spremnika

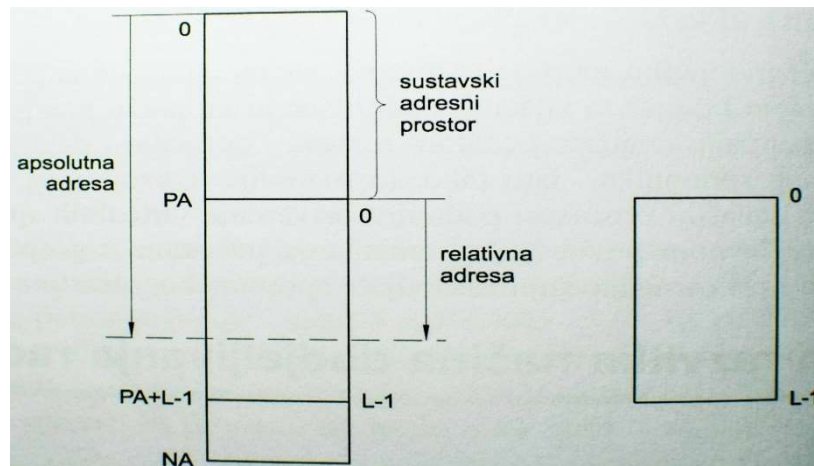
U najjednostavnijoj Von Neumannovoj arhitekturi računala unutar procesora neposredno se generiraju adrese koje se preko adresnih vodiča dovode do spremnika. Sve adrese koje izlaze iz procesora prolaze kroz adresni međuregistar. Te adrese:

- dolaze iz programskog brojila kada se iz radnog spremnika dohvaćaju instrukcije
- dolaze iz registra kazaljke stoga kada se adresira stog
- oblikuju se iz adresnog dijela instrukcija kada se adresiraju podaci

Kompilator prilikom prevođenja programa mora generirati niz strojnih instrukcija i poslagati ih jedne iza druge. U okviru nekih instrukcija morat će biti zapisane stvarne fizičke adrese. Ako zamislimo da takav strojni program moramo smjestiti u uzastopne bajtove spremnika, onda će u nekim bajtovima (odnosno nizu od četiri bajta ako je riječ od tridesetdvo-bitovnoj adresi) morati pisati brojevi koji su stvarne adrese u spremniku. Za takav program kažemo da je napisan u apsolutnom obliku. Apsolutne adrese, koje treba zapisati u takvom programu, ovise o tome gdje će program biti smješten u radnom spremniku kada bude pripremljen za izvođenje. Ako se predviđa da će prva adresa programa biti smještena na početnoj adresi PA, onda kompilator mora tu adresu uzeti u obzir pri određivanju apsolutnih adresa koje će se zapisati unutar strojnog programa. To znači da bi se početna adresa PA morala staviti na raspolaganje kompilatoru prije početka stvaranja strojnog programa. Međutim, prikladniji je drugi način pripreme programa, kod kojeg se prevođenje obavlja tako da se pretpostavi početna adresa programa jednaka nuli i generira strojni oblik u kojem su sve adrese

¹² Ibidem, str. 190

izračunate s obzirom na tu početnu nultu adresu. Te adrese zovemo relativnim adresama. Uz takav strojni oblik programa s relativnim adresama kompilator mora generirati posebne oznake uz one lokacije u kojima treba promijeniti adresu ovisno o stvarnoj početnoj adresi adresnog prostora u koji će program biti smješten. Veza između relativnih i apsolutnih adresa ilustrirana je slikom 10.



slika 10. Prevođenje relativnih adresa u apsolutne adrese pri statičkom dodjeljivanju spremnika

(Izvor: Budin L., G., op.cit., str 198)

U radnom je spremniku poželjno držati istodobno više programa. Oni moraju biti smješteni u različite dijelove spremnika s različitim početnim adresama. U povijesnom razvitku uspravljanja spremničkim prostorom jedan od zapaženih načina dodjele spremnika zasnivao se na podjeli spremnika na dijelove stalne veličine, tzv. stalne particije (engl. fixed partitions). Particije su mogle biti jednake veličine ili različitih veličina. Particije različitih veličina omogućavale subolje iskorištenje radnog spremnika jer su se apsolutni oblici manjih programa mogli prirediti za manje particije, a veći programi smjestiti u veće particije. Programi pripremljeni za jednu particiju nisu se jednostavno mogli preseliti u drugu particiju jer to seljenje zahtijeva novo generiranje apsolutnog oblika programa.

Štoviše, podjela korisničkog adresnog prostora na particije, uz programe koji su pripremljeni za izvođenje u toj particiji. Oni se nalaze na disku i mogu se po ukazanoj potrebi napuniti u radni spremnik. Ako ih se privremeno izbaci iz radnog spremnika, oni će se moći kasnije vratiti samo u istu particiju. Otuda dolazi i naziv statičko

raspoređivanje – programi se tijekom svog boravka u sustavu nalaze uvijek u istom dijelu radnog spremnika. Tijekom rada, spremnik neće biti potpuno iskorišten, i to iz dva razloga:

- Prvo, programi neće biti potpuno jednake veličine kao particije te će dijelovi particija ostati neiskorišteni. Taj način neiskorištenja dobio je naziv unutarnja fragmentacija (engl. internal fragmentation)
- Drugo, tijekom rada može se dogoditi da svi procesi čiji su programi smješteni u istu particiju bivaju blokirani pa ta particija radnog spremnika ostaje prazna. Pritom može postojati više procesa čiji programi čekaju na dodjelu radnog spremnika, ali oni ne mogu biti napunjeni u radni spremnik jer nisu pripremljeni za tu particiju. Taj način neiskorištenja spremnika dobio je naziv vanjska fragmentacija (engl. external fragmentation)

Naglasimo još i dodatno ograničenje da adresni prostori programa nisu mogli biti veći od najveće particije fizičkog spremnika.¹³

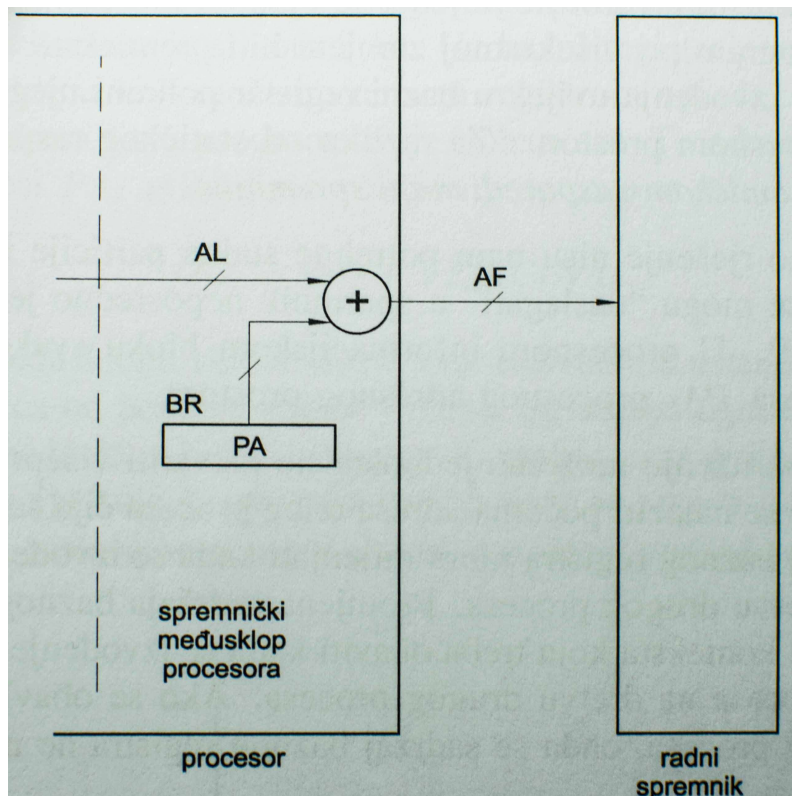
5.2.2. Dinamičko raspoređivanje radnog spremnika

S obzirom na to da je jedina operacija koju treba provoditi pri preračunavanju adresa pribrajanje početne adrese PA, nameće se pomisao da bi se to moglo obaviti sklopovski, i to uvijek kada se adresa iz procesora šalje prema spremniku. Ta se jednostavna zamisao može ostvariti uz sljedeće pretpostavke:

- Program će zauvijek ostati u obliku s relativnim adresama, tj. sve adrese unutar programa izračunate su kao da je početna adresa jednaka nuli. Taj adresni prostor nazvat ćemo logičkim adresnim prostorom
- Program će se smjestiti u radni spremnik u prostor koji započinje s početnom adresom PA. Taj ćemo prostor nazivati fizičkim adresnim prostorom
- U procesoru, na putu prema sabirnici, iza adresnog međuregistra treba dodati jedno sklopovsko zbrajalo i uz njega još jedan registar u koji će se zapisati početna adresa fizičkog adresnog prostora. Taj se registar naziva baznim registrom (engl. base register).

Ovakvo proširenje osnovnog modela prikazano je slikom 11:

¹³ Izvor: Budin L., G., op.cit., str. 197.



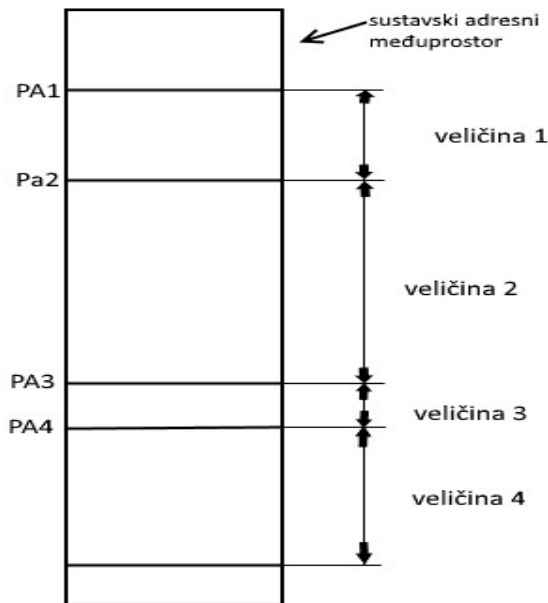
Slika 11. sklopovska pretvorba logičke adrese u fizičku adresu

(Izvor: ibidem str. 201)

Unutar procesora (u njegovim registrima) generira se logička adresa AL. Toj se adresi pribraja početna adresa PA koja prije početka izvođenja dretve mora biti pohranjena u bazni registar označen na slici s BR. Iz procesora prema radnom spremniku izlazi sada fizička adresa AF koja se dovodi do spremnika. Uočimo da se početna adresa pribraja svakoj adresi koja se u procesoru generira pa tako i adresama koje dolaze iz programskog brojila i registra kazaljke stoga. Time otpada briga o posebnom gledanju na adrese koje dolaze iz ta dva registra. Ovo sklopovsko proširenje procesora začetak je tzv. spremničkog međusklopa (engl. memory management unit – MMU) koji postaje sastavni dio svakog procesora.

Uz ovakvu sklopovsku potporu svi se programi priređuju tako da svaki program ima svoj logički adresni prostor koji smo za programe koji se izvode nazvali procesnim adresnim prostorom. Logički adresni prostor ne mijenja se tijekom izvođenja programa, bez obzira na mjesto kuda će program pri višekratnoj zamjeni biti premještan. Važno je samo da se prije početka njegova izvođenja uvijek u bazni registar pohrani njegova aktualna početna adresa u fizičkom adresnom prostoru. Za

razliku od statičkog raspoređivanja spremnika ovdje govorimo o dinamičkom raspoređivanju spremnika.



Slika 12. smještanje programa uz dinamično raspoređivanje radnog spremnika

(Izvor: izradio student)

Uz ovakvo sklopovsko rješenje nisu nam potrebne stalne particije i za njih pripremljeni programi. Programi se mogu „naslagati“ u spremnik neposredno jedan iza drugoga, kao što prikazuje slika 12. U procesnom informacijskom bloku svakog procesa mora biti zapisana početna adresa PA1 procesnog adresnog prostora.

U ovakvom radnom okruženju možemo jednostavno ostvariti višeprogramske rad. U baznom registru morat će se nalaziti početna adresa onog procesa čija se dretva upravo izvodi. To znači da se sadržaj baznog registra mora mijenjati kada se izvođenje prebacuje s dretve jednog procesa na dretvu drugog procesa. Promjena sadržaja baznog registra postaje tako sastavni dio promjene konteksta koju treba obaviti kada se izvođenje prebacuje s neke dretve unutar jednog procesa na dretvu drugog procesa. Ako se obavlja zamjena konteksta za dretve unutar istog procesa, onda se sadržaj baznog registra ne mijenja.

Pri ovakvom načinu uporabe radnog spremnika postoji velika opasnost da procesi jedan drugome na nedopušteni način zasmetaju. Naime, nikako se ne bi smjelo pretpostaviti da su svi programi uvijek potpuno ispravni i da će disciplinirano adresirati

samo unaprijed za njih predviđeni adresni prostor. U tom se radnom okruženju, uostalom, moraju izvoditi i programi koji se još razvijaju i ispituju i koji će s vrlo velikom vjerojatnošću pisati u prostor predviđen za druge procese. Pokazalo se da takve pogreške nije moguće izbjeći nikakvim programskim zaštitama i da treba potražiti neko sklopovsko razrješenje tog problema.

Osim toga, postoji problem i fragmentacije spremničkog prostora. Naime, tijekom vremena neki procesi bivaju blokirani i izbacuju se iz radnog spremnika pa tako također oslobađaju svoj procesni prostor kako bi načinili mjesta za programe procesa koji se mogu izvoditi. Ti će se procesi kasnije vratiti u radni spremnik, ali najvjerojatnije na neko drugo mjesto. S obzirom na to da veličine programa nisu jednake, u spremniku će se nakon nekog vremena između procesnih adresnih prostora pojaviti prazni dijelovi spremnika koje nazivamo rupama (engl. holes).

Spremnički međusklopovi proširivani su tako da se procesni adresni prostor može podijeliti na posebne dijelove, tzv. segmente:

- Segment u koji se pohranjuju instrukcije programa ili programski kod te se taj segment naziva kondim segmentom i adresira se iz programskog brojila
- Stogovni segment u koji se pohranjuje stog i adresira iz registra kazalje stoga
- Podatkovni segment u koji se pohranjuju podaci i čije adrese potječu iz adresnih dijelova instrukcija

Osnovni način suzbijanja fragmentacije svodi se na to da se rupe održavaju što većima kako bi se u njih mogli smjestiti novi programi. To se može postići tako da se:

- Pri svakom oslobađanju nekog procesnog prostora novonastala rupa spaja s eventualnim susjednim rupama u novu veću rupu
- Pri svakom novom zahtjevu za spremničkim prostorom potraži najmanja rupa u koju se može smjestiti novi program

Preostaje, nadalje, mogućnost da se u nekom trenutku, kada se ustanovi da je fragmentacija postala prevelika, privremeno obustavi izvođenje dretvi i „presloži“ programe u kompaktni prostor. Time se na kraju spremnika možda dobiva dovoljno velika rupa u koju se može smjestiti program koji čeka na dodjelu radnog spremnika.¹⁴

¹⁴ Izvor: Budin L., G., op.cit., str. 200.

5.3. Dodjeljivanje spremnika straničenjem

U različitim tipovima računalnih sustava susrećemo različite načine upravljanja spremnikom. U jednostavnijim se sustavima koriste metode koje se mogu ostvariti bez dodatnog sklopovlja namijenjenog upravljanju spremnikom. U drugim, koji imaju nekakvo sklopovlje za upravljanje spremnikom (npr. adresno zbrajalo spojeno između procesora i sabirnice), mogu se koristiti prikladne metode koje će iskoristiti to sklopovlje i poboljšati upravljanje spremnikom.

Složeni računalni sustavi, koji zahvaljujući razvoju računalne tehnologije danas uključuju i većinu ručnih računala, koriste postupak upravljanja spremnikom koji se naziva straničenje (engl. paging), a koji je omogućen složenim sklopovskim podsustavom za upravljanje spremnikom koje se nalazi na samom procesoru.

Upravljanje straničenjem, u osnovi algoritma, također koristi zbrajanje adresa. No u tom načinu upravljanja proces je podijeljen u dijelove istih veličina, koje nazivamo stranicama (engl. page). Svaka je stranica nezavisno spremljena u neki dio spremnika. Sustav za upravljanje će pri svakom zahtjevu morati odrediti kojoj stranici zahtjev pripada te gdje se ta stranica nalazi u spremniku. Nakon što se to utvrdi, iz drugog se dijela adrese određuje koliko je zahtjev udaljen od početka stranice te se taj broj zbraja na početnu adresu stranice u spremniku. Sve navedeno automatski obavlja sklopovlje pa je pretvorba adresa iznimno brza i ne uvodi kašnjenja u radu.

Radi jednostavnosti sklopovlja, veličina stranice je višekratnik broja dva: uobičajeno je to 4 kB (4096 okteta). Jedna će se takva stranica u spremnik spremiti na adresu koja je također isti višekratnik broja dva, tj. na adrese koje su višekratnik od 4 kB. Takve adrese imaju nižih p bitova (12 bitova, $2^{12} = 4096$) jednake nuli. Ovakav odabir omogućuje da se pri pretvorbi, nakon što se odredi početak spremnika gdje je stranica pohranjena (a što se naziva okvir, engl. frame), kao odmak od početka stranice jednostavno može iskoristiti nižih p bitova relativne adrese, tj. tih nižih p bitova ostaje nepromijenjeno, samo se redni broj stranice procesa mora pretvoriti u adresu početka te stranice, odnosno mjesta spremničkog prostora gdje je ona spremljena, što ćemo nazvati rednim brojem okvira spremnika. Za pretvorbu između rednog broja stranice procesa i rednog broja okvira spremnika koristi se tablica prevođenja dotičnog procesa u kojoj svaki redak opisuje po jednu stranicu. Prije pokretanja programa operacijski sustav mora pripremiti tablicu prevođenja za njega (i učitati dijelove programa u

spremnik). Pri izvođenju procesa pretvorba adresa koje on generira obavlja sklopovlje korištenjem pripremljene tablice. U stvarnim se sklopovskim ostvarenjima često koristi hijerarhijska organizacija tablice prevođenja prvenstveno radi smanjenja potrebnih dijelova tablice (npr. za manje procese).

Straničenje zahtijeva složeno sklopovlje, ali zato ima i mnoga dobra svojstva. Zaštita između procesa je besprijekorna, fragmentacija ne postoji, procesi mogu dinamički tražiti dodatne spremničke lokacije, čak se i program koji se cijeli ne može smjestiti u spremnik može izvoditi tako da se učitavaju samo oni dijelovi koji su trenutačno potrebni, a ostali su spremljeni na pomoćnom spremniku (disku). Sustav koji nema dovoljno spremničkog prostora (glavnog spremnika) može koristiti pomoćni spremnik da privremeno makne neke stranice koje trenutačno nisu potrebne, a da bi na njihovo mjesto smjestio one koje se traže. Naravno da će takve operacije u određenoj mjeri usporiti rad cijelog sustava jer je pomoćni spremnik znatno sporiji od glavnog, ali će ipak omogućiti pokretanje svih željenih programa. Zbog toga je pri odabiru potrebne veličine spremničkog prostora (pri kupovini računala) dobro poznavati zahtjeve koji će se na sustav postavljati i s motrišta spremničkog prostora. Premali glavni spremnik intenzivno će morati koristiti pomoćni spremnik i time znatno smanjiti učinkovitost sustava. S druge strane, preveliki će spremnik ostati neiskorišten (uzalud potrošen novac). Sustavi za rad u stvarnom vremenu, koji očekuju vrlo brzu reakciju na događaje u sustavu, ne koriste pomoćne spremnike jer oni unose značajna kašnjenja u rad sustava. Na primjer, ako se u obradi kritičnog događaja trebaju koristiti podatci koji se trenutačno nalaze u stranici koja je na pomoćnom spremniku, tada će proteći barem nekoliko milisekundi dok se ta stranica najprije ne dohvati s diska. Takva odgoda u nekim sustavima nije prihvatljiva te se u takvim sustavima ili straničenje uopće ne koristi, ili se koristi, ali bez pomoćnog spremnika, ili se koristi i pomoćni spremnik, ali samo za procese koji nisu vremenski kritični. Spremnik je znatno sporiji od procesora (barem nekoliko puta). Zato se koriste razne tehnološke metode da bi se taj nedostatak ublažio. Među najznačajnije spada korištenje priručnog spremnika samog procesora koji, iako je znatno manjeg kapaciteta, znatno ubrzava rad sustava. Brzina priručnog spremnika procesora bitno je veća od samog spremnika. Brži spremnici prilično su skuplji te se i sam priručni spremnik na procesoru dijeli na nekoliko razina različitih brzina, a da se takvom organizacijom postignu prihvatljive performanse uz prihvatljivu cijenu. Uobičajena podjela priručnih spremnika procesora je na tri razine (engl. cache

levels): prva (najmanja) razina (L1) koja radi na brzini samog procesora, druga (L2) koja radi nešto sporije, ali je većeg kapaciteta te treća (L3) koja je značajno većeg kapaciteta, ali je još sporija, međutim, ipak brža od glavnog spremnika. Zahtjevi za podacima u spremniku vrlo su često lokalizirani (vremenski bliski zahtjevi su i prostorno bliski – zahtjevi za susjednim lokacijama), procesor će vrlo često moći i samostalno raditi, korištenjem već dohvaćenih podataka u priručnim spremnicima, bez komunikacije s glavnim spremnikom. Veličina priručnog spremnika procesora prilagođena je njegovu korištenju. Dok će za zahtjevne poslužitelje trebati odabrati procesore sa znatnom količinom priručnog spremnika (zato i značajno skuplje), za obična su računala (kućna ili uredska) dovoljni i jednostavniji (jeftiniji) procesori s manjom količinom priručnog spremnika.

Korištenje priručnih spremnika mijenja i pristup spremniku. Iako program može zahtijevati samo jedan podatak, sklopovlje će dohvatiti cijeli blok podataka jer mu je to znatno isplativije iako će početno trajati nešto više od samo jednog zahtjeva (to će se višestruko isplatiti ako se i nekoliko kasnijih zahtjeva odnosi da dohvaćeni blok. Imajući na umu navedena svojstva glavnog i priručnih spremnika, pristup u izgradnji novih programa, koji su procesorski i spremnički zahtjevni, mora biti prikladan želi li se sklopovlje maksimalno iskoristiti. Korištenje priručnih spremnika je (najčešće) u potpunosti sklopovski riješeno. Poznavanjem svojstava i načina rada tog sklopovlja može se povećati učinkovitost programa, ali to nije nužno pri izradi nezahtjevnih programa. Matične ploče na koje se stavljaju sastavnice računalnog sustava obično i same podržavaju neke dodatne mogućnosti ubrzanja rada spremnika. S obzirom na ograničenja spremničkih modula koji se ugrađuju, a koji su određeni širinom sabirnice (brojem priključaka na ploču – brojem kontakata) i brzinom rada, matične ploče (tj. upravljački čipovi na njima), omogućuju uparivanje takvih modula radi postizanja efektivno šire sabirnice koja u istom razdoblju može prenijeti više podataka (npr. dvokanalno, trokanalno, engl. dual-channel, triplechannel).¹⁵

5.4. Novija kretanja memorijskih modula

S obzirom kako je tekao razvoj računala i svih njegovih komponenti, tako je tekao i razvoj memorijskih modula. Tvrdi disk, iako je zabilježio određen napredak kroz

¹⁵ http://www.zemris.fer.hr/~leonardo/os/dodatno/OS-skripta-za-srednje-skole/OS_uvod_1.pdf 25, 25. kolovoz 2016.

povijest, ostao je najslabija točka računala. S obzirom na svoja mehanička svojstva, tvrdi disk jednostavno nije mogao pratiti razvoj drugih komponenti računala te tako, u današnje vrijeme, mjesto primarnog diska za pohranu podataka preuzima SSD (Solid State Disk). Iako je hard disk i dalje nezaobilazna pojava u privatnoj upotrebi u kućanstvima (ponajviše zbog količine skladišnog prostora s obzirom na cijenu), u poslovnoj upotrebi situacija je potpuno suprotna i SSD preuzima vodstvo zbog znatno veće stabilnosti i brzine podataka s kojom se tvrdi disk ne može mjeriti.

5.4.1. SSD disk

SSD (eng. Solid state drive), su elektronički uređaji koji za razliku od tvrdog diska u sebi ne sadrže mehaničke pokretne dijelove koji vrše čitanje i zapisivanje podataka, već posjeduju NAND Flash memoriju postavljenu na pločici sklopa (PCB). Prednosti SSD tehnologije su mnogobrojne. Samom implementacijom NAND Flash memorije dobivamo statičan disk tj. nikakvih pokretnih dijelova unutar njega, a samim time manju razinu buke, otpornost na vibracije, veće brzine čitanja i zapisivanja te pristupno vrijeme (eng. access time). Primjer access time-a vidimo pri pokretanju programa i Windowsa.

Kao što je već navedeno u poglavlju 5.1., bitne karakteristike tvrdih diskova su trajanje traženja staze, rotacijsko kašnjenje i tzv. spin-up. Prva karakteristika je vrijeme koje je potrebno mehanizmu glave koja čita podatke da se pozicionira iznad staze u kojoj se nalazi traženi podatak. Rotacijsko kašnjenje se događa ako se traženi sektor na disku ne nalazi točno ispod glave u traženom trenutku pa treba pričekati dok okretom diska podatak ne dođe točno ispod glave. Ukupno ove dvije karakteristike uzimaju od 5 do 20 milisekundi, a uglavnom ovise o brzini okretanja diskova. Spin-up je vrijeme koje je potrebno da bi diskovi iz stanja mirovanja došli na potreban broj okretaja (primjerice 15.000 ili 7.200 okretaja u minuti), a to vrijeme može iznositi i nekoliko sekundi. Sreća je da se ovo događa samo prilikom uključivanja sustava. Sve ove karakteristike ne postoje na SSD-ovima, pa je pristup informacijama višestruko brži, gotovo trenutni i to od trenutka uključivanja uređaja.

Što se tiče veličine ovih jedinica, radi jednostavnosti je zadržana kompatibilnost s postojećim kućištima, pa su upotrebi dobro nam poznate oznake veličine (eng. Form factor) od 5.25, 3.5, 2.5 i 1.8 inča. Malo je poznato da to nisu fizičke dimenzije cijelog uređaja nego dimenzije rotirajućih ploča u klasičnim tvrdim diskovima. U osobnim i

prijenosnim računalima najčešće se upotrebljavaju jedinice veličine 2.5 inča. Postoje još i jedinice koje su izgledom i veličinom vrlo slične radnoj memoriji za osobna i prijenosna računala (eng. Base-board factors).

Kapacitetom SSD-ovi još nisu dostigli klasične diskove. Brojke se danas kreću se od 50GB do 256GB, a rijetke su i preskupe jedinice većeg kapaciteta. Međutim, proizvođači pokušavaju dobiti na kapacitetu novim kontrolerima koji logički spajaju više diskova u jedan, ali ta rješenja višestruko dižu ionako visoku cijenu. Iz svih ovih razloga u osobna se računala obično stavlja jedan SSD na kojem se nalazi operativni sustav i programi, a podaci se drže na "običnim" diskovima. Međutim, u poslužiteljima i visokodostupnim sustavima je druga priča, tamo se zbog brzine sve više koriste isključivo SSD jedinice.

Sučelja kojima se SSD spaja na računalo su ista kao i za klasične tvrde diskove. Kako su neka od njih već zastarjela, sad se uglavnom mogu naći Serial ATA (SATA), Serial attached SCSI (SAS), PCI Express i USB sučelja.

Najvažniji dio SSD-a je njegov kontroler. O njemu ovise vrsta sučelja i, još važnije, performanse same jedinice. Neke od funkcija koje kontroler obavlja su ispravljanje grešaka (Error correction – EEC), mapiranje loših blokova (Bad block mapping), keširanje čitanja i pisanja (Read-Write caching), te enkripcija. Upravo se u kontrolerima kriju najveće razlike među danas dostupnim SSD-ovima. Neki proizvođači dizajniraju svoje kontrolere, a drugi koriste već postojeće, koje nastoje poboljšati sitnim izmjenama u kontrolnom softveru (eng. Firmware). Neki od poznatijih kontrolera su SandForce (koriste ga, između ostalih, Kingston, Intel te SanDisk), IndiLinx (OCZ), Link A Media (Corsair) i MDX (Samsung). Već ste zaključili da većinu proizvođača poznajete kao proizvođače radne memorije, a razlog je više nego očit, u pitanju je slična tehnologija.¹⁶

5.4.2. *Nand flash memorija*

NAND flash nam je dobro poznat jer se koristi u memorijskim karticama i USB flash jedinicama. Dvije velike prednosti flash memorije su pamćenje informacija bez konstantnog izvora energije (eng. non-volatile flash memory) i niža cijena. Upravo su to razlozi zbog kojih su ovi SSD-ovi dominantni na tržištu osobnih računala. Ipak,

¹⁶ <https://sysportal.carnet.hr/node/1211>, 26. kolovoz. 2016

NAND flash ima mana. Kod klasičnih tvrdih diskova stari se podatak jednostavno može "prepisati" novim. Da bi se na flash memoriji upisao novi podatak, najprije treba izbrisati stari. Ovaj proces produžuje operaciju "piši". Pokušaj rješenja ovog problema doveo je do uvođenja naredbe TRIM kojom operativni sustav javlja kontroleru koji blokovi više nisu u upotrebi, pa njih obriše sam kontroler kad je uređaj u stanju mirovanja (eng. idle). Ova je naredba implementirana u svim novijim operativnim sustavima (od 2008. godine). Nadalje, flash memorija ima svoj rok trajanja. Za to su "krivi" ciklusi piši/briši koji uzrokuju fizičku degradaciju oksidnog sloja (izolatora) kojim su obloženi logički elementi. Da bi se ovaj proces usporio, koristi se tzv. Wear leveling metoda, a to su algoritmi ugrađeni u kontroler koji ravnomjerno raspoređuju cikluse pisanja/brisanja po cijelom SSD uređaju. Zbog degradacije sklopova defragmentacija SSD-ova nije preporučljiva. Srećom ona nije ni potrebna zbog brzine pristupa informacijama.

Jedan način mjerenja performansi SSD jedinica (i tvrdih diskova) jest brzina transfera podataka pri čitanju tj. pisanju. Za klasične tvrde diskove ona dostiže 140MB/s, a NAND flash jedinice postižu brzine do 600MB/s. Mnogo je važniji faktor tzv. IOPS (eng. Input/Output operations Per Second) - broj operacija u sekundi koji jedinica može izvesti. Kao i kod većine stvari, postoji podjela na tzv. Enterprise i Consumer NAND flash SSD jedinice. Kod osobnih računala operacija čitanja informacija brojkom daleko nadmašuje operaciju pisanja. Zato su Consumer jedinice optimizirane za mnogo čitanja i manje pisanja, a broj piši/briši ciklusa kreće se do 10.000, što daje otprilike 5-7 godina "normalnog" rada. S druge strane, Enterprise jedinice se upotrebljavaju u okolišu u kojem je otprilike jednak broj čitanja i pisanja u jako dugom periodu (npr. aplikacije na poslužiteljima). Zbog toga je konstrukcija i testiranje Enterprise jedinica mnogo rigoroznija, a broj piši/briši ciklusa doseže i do 100.000. NAND flash SSD-ovi postižu do 500 IOPS-a, a latencija iznosi otprilike 25 mikrosekundi naspram 2 milisekunde za klasične (vrhunske) diskove koji se "vrte" na 15.000 okretaja u minuti. Broj IOPS-a se za flash jedinice kreće od 500 do 50.000, što je opet nadmoćno u odnosu na klasične tvrde diskove (do 1.200). Naravno, razvojem tehnologije flash SSD-ovi postaju sve brži i pouzdaniji pa se broj ciklusa piši/briši stalno povećava.

Druga vrsta memorije koja se koristi za izradu SSD-ova je ustvari klasična radna memorija. Riječ je o DRAM-u koji za pamćenje informacija zahtijeva stalan izvor napajanja, obično putem ugrađene baterije. Znamo da su memorije sve brže pa je tako i ova vrsta SSD-ova prilično brža od one bazirane na flash memoriji, ali je i skuplja.

Međutim, ova vrsta memorije ne pati od degradacije sklopova s vremenom, što znači da joj je vijek trajanja duži. DRAM jedinice su nadmoćne u broju IOPS-a koji se kreće do 200.000, a latencija iznosi do maksimalnih 20 nanosekundi. Što se tiče brzine transfera, brojke su slične SSD-ovima baziranim na flash memoriji, ali ova karakteristika nije toliko bitna. Ona je ograničena sučeljem kojim se SSD-ovi spajaju na sustav. Primjerice, SATA sučelje revizije 3 ima limit od 600MB/s, a njega su SSD-ovi već dostigli. Kako DDR3 ima teoretski limit od 6400MB/s, možemo očekivati više vrijednosti brzine transfera DRAM SSD-ova. Zahvaljujući svojim karakteristikama, DRAM SSD-ovi se koriste u poslužiteljima i visokodostupnim sustavima, a budući su takva mjesta opremljena dodatnim izvorima napajanja, činjenica da im treba stalno napajanje nije presudna kod odabira.

Jedan od većih nedostataka SSD diskova je problem s gubitkom podataka. Klasični tvrdi disk se u laboratorijskim uvjetima mogao rastaviti te su se podaci mogli vratiti. Vraćanje podataka sa SSD-a je teži posao. Ako jedinica nije fizički oštećena i bazirana je na flash memoriji, moguće je vratiti podatke pomoću specijaliziranih programa. Međutim, za gubitak podataka na DDR nema pomoći ukoliko se dogodi nestanak struje. Zato su sigurnosni sustavi kompliciraniji i skuplji – za sigurnosne kopije se koriste ili klasični tvrdi diskovi ili flash jedinice. Performanse flash SSD-ova loše su u radu s malim datotekama. Kako su blokovi flash memorije veliki 4KB, nemoguće je pročitati ili zapisati manje od toga, čak i kad je zapis manji. Zato su u radu s malim datotekama klasični tvrdi diskovi još uvijek nadmoćniji. DRAM SSD-ovi ne pate od ovih boljki. Međutim, intenzivno se razvijaju datotečni sustavi prilagođeni flash memoriji koji će omogućavati još veće brzine prijenosa i zapisivat će male blokove. Jedan od njih je Universal Flash Storage (UFS) koji se već koristi na digitalnim fotoaparatom, memorijskim karticama i mobilnim uređajima. Unatoč nekim od nedostataka, SSD diskovi predstavljaju budućnost te se očekuje kako će s vremenom istisnuti s tržišta klasične tvrde diskove.¹⁷

¹⁷ <https://sysportal.carnet.hr/node/1211>, 26. kolovoz 2016.

6. Datotečni podsustav

Svi sadržaji koji se u računalu trebaju trajno čuvati pohranjuju se u datoteke. Praktički svi poslovi koje izvodimo računalom oslonjeni su na uporabu i pretvorbu datoteka (engl. file). Zbog toga se korisnik već u svom prvom pristupu računalu susreće s datotekama. Taj se pristup obavlja posredstvom dijela operacijskog sustava koji nazivamo datotečnim sustavom (engl. file system).

S obzirom na to da se sadržaji radnog spremnika gube pri svakom isključivanju računala, datoteke se čuvaju na vanjskim spremnicima. Najčešće korišteni vanjski spremnici su diskovi. Na svaki se disk, osim samih datoteka, pohranjuju i tablice ili direktoriji (engl. directories) u koje se smještaju imena datoteka s kazaljka koje omogućuju pristup do datoteka. Tablice se uobičajeno organiziraju po odjeljcima i pododjeljcima (engl. sub-directory) kako bi se olakšalo pronalaženje datoteka. Takve se tablice mogu slikovno prikazati kao razgranato stablo u čijoj se posljednjoj razini nalaze nazivi datoteka.

Za svaku se datoteku svi važni podaci, tzv. atributi datoteke (engl. attributes) smještaju u jedan zapis koji nazivamo opisnikom datoteke (engl. file descriptor). Podatke iz tog zapisa koristi operacijski sustav, a ne korisnički programi. Takav smještaj opisnika omogućuje da se ista datoteka može nazvati različitim imenima i da se do nje može pristupiti iz različitih ogranaka direktorija.

Datoteke uz svoj naziv uobičajeno dobivaju sufiks (odvojen od imena točkom) kojim se opisuje tip datoteke. Datoteke mogu sadržavati primjerice:

- Programe u strojnom obliku pripremljenom za izvođenje (uobičajeno sa sufiksom .exe)
- Izvorne programe napisane u višem programskom jeziku (uobičajeno sa sufiksom koji ukazuje na jezik: c, cpp, pas i sl.)
- Datoteke koje sadrže tekstove i dokumente (uobičajeno sa sufiksima txt, asc ili doc).

Datotečni sustav omogućuje provođenje različitih operacija nad datotekama. Za određeni tip datoteka dopuštene su samo određene operacije, tako se, primjerice,

datotekom koja sadrži program pripremljen za izvođenje može obaviti samo operacija pokretanja programa.

Znakovna se datoteka ne može pokrenuti kao program (čak i ako je u datoteci pohranjen program napisan nekim višim programskim jezikom!).

Za pojedine tipove datoteka može se odrediti dopuštene načine njihove uporabe (samo izvođenje, samo čitanje, samo pisanje). Općenito, pristup do određenih datoteka može se sigurnosno zaštititi, tj. dopustiti ga samo za to ovlaštenim korisnicima.

Datoteke, osim trajnog pohranjivanja nekih sadržaja, imaju još jednu važnu svrhu u računalnim sustavima: one služe i za razmjenu informacija između programa. Datoteka može biti stvorena i popunjena od strane jednog programa i nakon toga njezin sadržaj može dohvatiti drugi program.

Prema tome, može se reći da datoteke u računalnim sustavima imaju dvojaku ulogu:

- One služe za trajno pohranjivanje svih oblika sadržaja
- S pomoću njih se može obavljati razmjena informacija između različitih programa¹⁸

6.1. Struktura datoteka

Datotečni sustav u načelu se ne brine o sadržaju datoteka. Ipak je, zbog potpunijeg razumijevanja datotečnih operacija, prikladno razmotriti neke od mogućih načina strukturiranja datoteka.

Binarna datoteka

Najjednostavniji oblik datoteke koristi se za pohranjivanje strojnih programa pripravnih za izvođenje. Takva se datoteka može promatrati kao niz bitova (pohranjenih u bajtove) koji se kao cjelina prenose iz jednog spremnika u drugi, i to bez ikakvih promjena.

Budin L., G., op.cit., str. 239.

Nestrukturirana datoteka

U današnjim se operacijskim sustavima pretežito upotrebljavaju datoteke koje se sastoje od niza bajtova i nemaju nikakvu strukturu. Datotečni sustav omogućuje pristup do pojedinog bajta s pomoću datotečne kazaljke (engl. file pointer). Pri otvaranju datoteke datotečna kazaljka postavlja se na početak datoteke. Datotečni sustav svojim funkcijama omogućuje čitanje niza bajtova iz datoteke i pisanje niza bajtova u datoteku. Datotečna se kazaljka pri čitanju i pisanju pomiče za broj pročitanih, odnosno upisanih bajtova. Prilikom čitanja, funkcije datotečnog podsustava posebnom oznakom obilježavaju kraj datoteke (engl. End of File), čime se može ispitati uvjet dolaska do kraja datoteke. Pokušaj čitanja iza kraja datoteke izaziva pogrešku, dok pisanje u datoteku može izazvati produljenje datoteke i pomicanje oznake novog kraja datoteke.

Oponašanje indeksne strukture

U nekim ranijim operacijskim sustavima datotečni je sustav podržavao i operacije s tzv. indeksnim datotekama. Danas se uporaba indeksno organiziranih struktura zapisa prepušta sustavima baza podataka. Programski sustavi za održavanje baza podataka omogućavaju pohranjivanje i dohvaćanje zapisa usklađeno s vrijednostima pojedinih polja tih zapisa. Međutim, indeksna se struktura može organizirati unutar korisničkog programa uporabom tablica za pohranjivanje parametara strukture te izgradnjom funkcija za pristupanje do pojedinih elemenata strukture uporabom postojećih API funkcija za pristupanje nestrukturiranoj datoteci.¹⁹

¹⁹ Budin L., G., op.cit., str. 240

7. Višediskovni zalihosni spremnici

Napredak svojstava magnetskih diskova mnogo je sporiji jer on ovisi o napretku elektromehaničkih svojstava diskovnih naprava. Stoga diskovlje postaje ograničavajući faktor u sveopćem unapređivanju računalnih sustava. Dobavljanje nekog sadržaja s diska ili njegova pohranjivanja ovisi i o njegovu smještanju u sektore diska. Kompaktni smještaj pri kojem se minimizira trajanje premještanja glava s cilindra na cilindar omogućuje maksimalno iskorištenje širine pojasa pristupa disku. Raspršeni smještaj blokova podataka (koji minimizira fragmentaciju diska) smanjit će brzinu prijenosa jer se prijenos ne može obavljati u razdoblju premještanja glava i rotacijskog kašnjenja.

Upravljački sklopovi, koji su po brzini rada usporedivi s procesorom, nezamjetno utječu na brzinu prijenosa. Štoviše, upravljački sklopovi mogu poslužiti za neka poboljšanja u prijenosu više uzastopnih blokova jer se njihovi spremnici mogu upotrijebiti kao priručni spremnici.

Ubrzanje pri prijenosu može se, primjerice, postići tako da se:

- Pri čitanju jednog bloka u spremnik upravljačkog sklopa prenese sadržaj cijele staze i tako ubrza eventualno čitanje nekog drugog bloka s iste staze
- Posluživanje više zahtjeva koji su prispjeli u upravljački sklop ne obavlja redom prispjeća već onim redoslijedom koji minimizira pomake glava

Međutim, sva takva nastojanja mogu ubrzati prijenos samo do njegova maksimalnog iznosa određenog elektromehaničkim svojstvima diskova. To se ograničenje može zaobići smještanjem podataka na više diskova kojima se može istodobno pristupati. Višediskovni se sustav sastoji od polja diskova (engl. disk array) kojima se pristupa paralelno. Podaci se na diskove mogu raspoređivati na različite načine. Najjednostavnije je promatrati diskove kao nezavisne cjeline i svaki od njih adresirati nezavisno. Adresni prostor diska čine adrese sektora pri čemu je adresa određena rednim brojem cilindra, rednim brojem staze u cilindru i rednim brojem sektora na stazi.

U drugom se načinu raspoređivanja cijelo polje diskova promatra kao jedan logički adresni prostor. Pritom je osnovna jedinica adresiranja podatkovni pojas ili, kraće,

samo: pojas (engl. dana stripe, stripe), koji se na diskove smješta kružnim (engl. round robin) adresiranjem.

Razlikujemo dvije vrste pojasne organizacije:

- Sitno zrnatu pojasnu organizaciju (engl. fine-grained striping)
- Krupno zrnatu pojasnu organizaciju (engl. coarse-grained striping).

Uporaba većeg broja diskova povećava vjerojatnost pojave kvarova. Zbog toga se pri izgradnji višediskovnih sustava nužno nameće uvođenje stanovite zalihosti i tehnika za oporavak od kvarova. Uporaba zalihosnih diskova za povećanje pouzdanosti unosi potrebu za dodatnim pohranjivanjima i dobavljanjima podataka, što smanjuje efektivnu brzinu prijenosa. Prema tome, postizanje veće brzine i veće pouzdanosti dva su kontradiktorna zahtjeva.

8. Sigurnost računalnih sustava

Sigurnost računalnih sustava postaje sve važnija jer u današnjem svijetu sve više korisnika na sve više načina koristi sve više informacija koje su raspršene u raspodijeljenim sustavima. U takvom svijetu postoji sve veća opasnost od neovlaštene uporabe informacija, podmetanja krivih informacija ili uništavanje informacija. Opća koncepcija sigurnosti ima svoje moralne i pravne aspekte koji se reguliraju zakonodavstvom i odgovarajućim kaznenim mjerama. Ugrožavanje sigurnosti računalnih sustava moguće je klasificirati na različite načine. Jedna od mogućih podjela sigurnosnih mehanizama je sljedeća:

- zaštita od vanjskih utjecaja
- zaštita ostvarena sučeljem prema korisniku
- unutarnji zaštitni mehanizmi
- komunikacijski zaštitni mehanizmi

8.1. Vrste napada na sigurnost

Vrste napada na sigurnost mogu se najbolje ilustrirati modelom u kojem je izvorište informacija povezano s odredištem komunikacijskim kanalom. Izvorište i odredište mogu se nalaziti:

- u jednom računalu (primjerice, pri razmjeni sadržaja između radnog i vanjskog spremnika ili pri prenošenju podataka iz jedne datoteke u drugu)
- u različitim računalima raspodijeljenog sustava (pri čemu su u komunikacijski kanal uključene telekomunikacijske naprave)

Načini napada na sigurnost:

- Prisluškivanje
- Prekidanje
- Promjena sadržaja poruka
- Izmišljanje poruka
- Lažno predstavljanje
- Poricanje

Na temelju opisanih mogućih napada na sigurnost proizlazi da se sigurnost računalnih sustava zasniva na ispunjavanju šest osnovnih sigurnosnih zahtjeva. To su: povjerljivost, raspoloživost, bespriječnost, autentičnost, autorizacija, neporecivost.

8.2. Osnove kriptografije

S obzirom na veliku odgovornost koju nose računalni sustavi, vidljivo je kolika je važnost uspostave sigurnosnih mehanizama u umreženim računalnim sustavima. Moglo bi se reći da mnoga korisna ostvarenja uporabe računalnih mreža u praktički svim područjima ljudske djelatnosti dobrim dijelom ovise o razvitku pouzdanih zaštitnih mehanizama koji će osigurati primjerenu sigurnost sustava. Pokazat će se da se svi sigurnosti zahtjevi, osim raspoloživosti, mogu zadovoljiti uvođenjem kriptiranja sadržaja koji će razmjenjivati u umreženim računalnim sustavima. S obzirom na to da je u komunikacijskom sustavu nemoguće spriječiti prisluškivanje podataka, pokazalo se razumnim načiniti podatke nerazumljivima neovlaštenim uljezima. Podaci koji u svom izvornom obliku predstavljaju neku korisnu informaciju mogu se postupkom kriptiranja prevesti u oblik u kojem se ta informacija više ne prepoznaje.

8.2.1. Kriptiranje

Budući da su počeci kriptiranja povezani s prenošenjem pisanih informacija u obliku tekstova, u kriptografskoj se terminologiji izborni oblik podataka naziva razgovjetnim ili jasnim tekstom (engl. plaintext, cleartext).

Postupkom kriptiranja (engl. encryption, enciphering) jasni tekst se prevodi u kriptirani tekst (engl. ciphertext). Obrnuti postupak prevođenja kriptiranog teksta u jasni tekst naziva se dekriptiranjem (engl. decryption, deciphering). U današnje vrijeme, kada nam na raspolaganju stoje vrlo moćna računala, ne mogu se primjenjivati naivne stare metode kriptiranja koje su se, u načelu, zasnivale na zamjeni znakova prema nekim složenim pravilima. Današnji kriptografski postupci su parametarske matematičke funkcije, odnosno algoritmi kojima se nizovi bitova jasnog teksta preračunavaju u nizove bitova kriptiranog teksta i obrnuto.

Razlikujemo:

- Simetrične kriptosustave (DES, IDEA, AES)
- Asimetrične kriptosustave (RSA)

Suvremeni operacijski sustavi kao svoj sastavni dio nude vrlo dobra rješenja autentifikacije i autorizacije te se u sučeljima prema korisničkim programima kroz API (Application Programming Module) funkcije nude prikladni sigurnosni mehanizmi. Novija radna okruženja pojedinih programskih jezika nadopunjuju se također postupcima koji se mogu ugrađivati u primjenske programe (tako se, primjerice, za programski jezik Java pronalazi API funkcije za ostvarenje digitalnog potpisivanjam izradu sažetaka poruka, raspodjelu ključeva i autorizacije).

Za raspodijeljena okruženja u širokoj je uporabi protokol Secure Sockets Layer – SSL (izvorno razvijen u tvrtki Netscape) koji djeluje kao međurazina između TCP-a i primjenske razine. On omogućuje odabir različitih kriptosustava za ostvarenje pojedinih sigurnosnih mehanizama.²⁰

²⁰ Ibidem, str.280.

9. Zaključak

Operativni sustavi predstavljaju ključnu vezu između korisnika i računala – omogućuju komunikaciju jedne strane s drugom. Osim toga, njihova zadaća je osigurati stabilan rad računala u vidu optimalne potrošnje energije, upravljanja memorijom, izvršavanje odgovarajućih programa s obzirom na prioritet i slično.

Budući da memorijski aspekt računalnog sustava omogućava računalu rad s podacima te isto tako omogućava računalu pohranu podataka i odgodu korištenja podataka, jasno je vidljiva njegova važnost. S obzirom na veliku količinu podataka koja se koristi tijekom rada računala te više vrsta memorije, postoji problem upravljanja memorijom. Na operativnom sustavu je – po napatku korisnika računala – da upravlja tim podacima, određuje koji memorijski modul što preuzima te što se u koju memoriju sprema.

Moderni operativni sustavi donose velike mogućnosti. Konstantna izvedba više programa u realnom vremenu donosi nove zahtjeve za operativne sustave. Operativni sustav ima zadatak alokacije raspoložive memorije kako bi se svi programi mogli izvoditi u pozadini, a da pri tome računalo korisnika i dalje ima raspoložive memorije za nove zadatke ukoliko se korisnik računala za njih odluči.

Slijedom navedenog, vidljivo je da su operativni sustavi najvažniji, ali i najsloženiji softver na računalu. Imaju višestruku funkcionalnost, a bez njih rad na računalu nije moguć za krajnjeg korisnika. To je ujedno i razlog zašto je proizvođača (kvalitetnih) operativnih sustava jako malo te su neke od inačica operativnih sustava dosta skupe. U kontekstu krajnjeg korisnika, operativni sustav predstavlja grafičko sučelje putem kojeg on upravlja računalom. Pozadinski procesi mu nisu vidljivi i nije opterećen dodatnim informacijama – operativni sustav obavlja taj posao za njega. Pitanje izbora operativnog sustava u konačnici se svodi na preferencije krajnjeg korisnika budući da svaki od operativnih sustava na tržištu nudi nešto drugo, odnosno nešto drugačiji pristup upravljanju računalom.

10. Literatura

a) Knjige

1. Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010.
2. Silbershatz A., Galvin P.B., Gagne G.: Operating system concepts with Java, sixth edition, USA 2004.

b) Internet:

http://www.teach-ict.com/gcse_computing/ocr/211_hardware_software/computer_system/miniweb/pg2.htm, 15.8.2016

<http://passnownow.com/classwork-exercise-and-series-computer-jss-2-the-computer-system/>, 15.8.2016

<http://flylib.com/books/en/2.944.1.11/1> 15.8.2016

http://www.webopedia.com/TERM/O/operating_system.html, 19.8.2016

<https://loomen.carnet.hr/mod/book/view.php?id=181686&chapterid=41751>, 19.8.2016

http://www.tutorialspoint.com/operating_system/os_memory_management.htm 22.8.2016

<https://sysportal.carnet.hr/node/1211>, 26.8.2016

http://www.zemris.fer.hr/~leonardo/os/dodatno/OS-skripta-za-srednje-skole/OS_uvod_1.pdf
25.8.2016

11. Popis slika

Slika 1: Operativni sustav (OS) dijagram.....	6
Slika 2: Primjer računalnog sustava.....	7
Slika 3: Operativni sustav (OS) dijagram.....	8
Slika 4: Funkcijski Von Neumannov model računala.....	10
Slika 5: Sklopovlje računala povezano sabirnicom.....	11
Slika 6: Višedretveni rad na višeprocessorskom računalu.....	12
Slika 7: Višedretveni rad na jednoprocessorskom računalu.....	13
Slika 8: Spajanje ulazno-izlaznih naprava na sabirnicu računala.....	14
Slika 9: Mehanički izgled diska.....	18
Slika 10: Prevođenje relativnih adresa u apsolutne adrese.....	19
Slika 11: Sklopovska pretvorba logičke adrese u fizičku adresu.....	21
Slika 12: Smještanje programa uz dinamično raspoređivanje radnog spremnika.....	22

12. Sažetak

Računala sama po sebi beskorisna su u komercijalnoj upotrebi bez operativnog sustava. Također, osim operativnog sustava, računalo se sastoji od nekoliko dijelova koji zajedno čine funkcionalnu cjelinu koja se naziva računalni sustav.

Operativni sustavi povezuju sve te dijelove te omogućuju jednostavan pristup čovjeka računalu, odnosno omogućava komunikaciju čovjeka i računala putem grafičkog sučelja. Prvi operacijski sustavi razvili su se 50-ih godina 20. stoljeća, a operativni sustavi kakve danas poznajemo razvili su se 80-ih godina 20. stoljeća te njihov razvoj traje još i danas, a najpoznatiji su Windows, Linux, Mac OS.

Današnje računalo zasniva se na koncepcijskom modelu kojeg je 1945. godine opisao John Von Neumann, a prema tome modelu računalo se sastoji od ulaznog i izlaznog dijela, radnog spremnika, aritmetičko-logičke jedinice i upravljačke jedinice.

Upravljanje memorijom funkcija je operativnog sustava koja upravlja primarnom memorijom i premješta procese naprijed natrag između glavne memorije i diska između izvođenja. Upravljanje memorijom prati svaku memorijsku lokaciju, neovisno o tome je li ona dodijeljena nekom procesu ili je slobodna. Upravljanje memorijom odlučuje koji proces će dobiti memoriju i u koje vrijeme te prati memoriju ne bi li ostala slobodna ili nedodijeljena te sukladno tome joj ažurira status. Postoji radna memorija te trajna memorija (tvrđi disk).

Cilj rada je objasniti rad operativnih sustava unutar računalnog sustava te pobliže objasniti funkciju upravljanja memorijom unutar operativnog sustava. Detaljno su opisani principi osnovnog modela računala kako bi se postavio temelj za detaljnije analiziranje problematike upravljanja memorijom. Najvažnija poglavlja su: upravljanje memorijom, datotečni podsustav te višediskovni zalihosni spremnici.

13. Summary

Computers in commercial use are useless without an operating system. Also, besides an operating system, a computer has a few parts which together make one unit called a computer system.

An operating system connects all those parts together and allows for easy access to the computer, and allows communication between the user and the computer through the graphical interface. The first operating systems were developed during the 1950s, and the development of operating systems that we know today began in the 1980s and continues to this day – the most popular ones are Windows, Linux and Mac OS.

Today's computers are based on the concept model described by John Von Neumann in 1945. Like the '45 model, a modern computer consists of input units and output units, an arithmetically-logical unit and a control unit.

Controlling the memory is a function of the operating system that controls the primary memory and transfers processes back and forth between the main memory and the disk between execution. Controlling the memory tracks every memory location, regardless of it being used by a process or not. Controlling the memory determines which process will use the memory and at what time. It also tracks the memory to check whether it is used or not and accordingly update its status. Two memory types exist: Random Access Memory (RAM) and permanent memory.

The goal of the paper is to explain how an operating system works inside a computer and give a detailed explanation of controlling the memory inside the operating system. The basic principles of the initial computer model from 1945 are described in detail to set ground for more detailed analysis of problems with controlling the memory. The most important chapters are: controlling the memory, file subsystem, and multi-disk redundant containers.