Operacijski sustavi na računalima

Klanac, Ivan

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:574262

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

University North Digital Repository

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN



DIPLOMSKI RAD br. 039-MMD-2021

OPERACIJSKI SUSTAVI NA RAČUNALIMA

DIPLOMSKI RAD

Ivan Klanac, 1249/336D

Varaždin, rujan 2021.

SVEUČILIŠTE SJEVER

SVEUČILIŠNI CENTAR VARAŽDIN

Studij Multimedija



OPERACIJSKI SUSTAVI NA RAČUNALIMA

Student

Ivan Klanac – 1249/336D

Mentor

doc. dr. sc. Robert Geček

Varaždin, rujan 2021.

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ diplomski sveučilišni studij Multimedija

PRISTUPNIK Ivan Klanac | JMBAG 0016121268

DATUM 31.08.2021. | KOLEGIJ Projektni studio 1

NASLOV RADA Operacijski sustavi na računalima

NASLOV RADA NA ENGL. JEZIKU Operating systems on computers

MENTOR Robert Geček | ZVANJE doc.art. dr.sc.

ČLANOVI POVJERENSTVA

1. doc.dr.sc. Andrija Bernik - predsjednik
2. doc.dr.sc. Domagoj Frank - član
3. doc.art.dr.sc. Robert Geček - mentor
4. izv.prof. dr.sc. Emil Dumić - zamjenski član
5.

Zadatak diplomskog rada

BROJ 039-MMD-2021

OPIS

Ovaj diplomski rad prikazuje i opisuje kako su se operacijski sustavi razvijali kroz povijest te je detaljno opisan rad operacijskog sustava u suštini. Navedeni su i objašnjeni razno razni zadaci koje rade operacijski sustavi. Istraženo je koje hardverske komponente trebaju imati računala za rad na pojedinim operacijskim sustavima. Detaljnije je opisan rad programa, procesa, dretvi, jezgra operacijskog sustava, gospodarenje spremničkim prostorom i na kraju datotečni podsustav. Proučit će se gdje se sve koriste i gdje se sve mogu primjenjivati operacijski sustavi i na koji način oni pripomažu i koje pogodnosti imaju za krajnjeg korisnika.

ZADATAK URUČEN 07.09.2021. | POTPIS MENTORA

REPUBLIKA HRVATSKA SVEUČILIŠTE SJEVER KOPRIVNICA

PREDGOVOR

Ovim putem zahvaljujem se svojem mentoru doc. dr. sc. Robertu Gečeku koji mi je svojim savjetima, uputama i sugeriranjem olakšao izradu diplomskog rada.

Ivan Klanac

Sažetak

Ovaj diplomski rad prikazuje i opisuje kako su se operacijski sustavi razvijali kroz povijest te je detaljno opisan rad operacijskog sustava u suštini. Navedeni su i objašnjeni razno razni zadaci koje rade operacijski sustavi. Istraženo je koje hardverske komponente trebaju imati računala za rad na pojedinim operacijskim sustavima. Detaljnije je opisan rad programa, procesa, dretvi, jezgra operacijskog sustava, gospodarenje spremničkim prostorom i na kraju datotečni podsustav. Proučit će se gdje se sve koriste i gdje se sve mogu primjenjivati operacijski sustavi i na koji način oni pripomažu i koje pogodnosti imaju za krajnjeg korisnika.

Ključni pojmovi: tehnologija, operacijski sustav, uređaji, internet, primjena

Summary

This dissertation shows how operating systems have evolved through history and it describe in detail the operation of the operating system in essence. The various tasks performed by operating systems are listed and explained. It was investigated which hardware components computers should have so they can work on individual operating system. The operation of the program, process, threads, operating system kernel, container space management and finally the file subsystem are described in more detail. It will examine where everything is used and where all operating systems can be applied and how they can help and what benefits they have for the end user.

Key words: technology, operating system, devices, internet, appliance

Sadržaj

1. Uvod 1
2. Metode i tehnike rada 2
3. Podjela operacijskih sustava 3
3.1. Podjela operacijskih sustava s obzirom na tip koda 3
3.2. Podjela operacijskih sustava s obzirom na izvođenje dretvi 3
3.3. Tipovi operacijskih sustava 3
3.3.1. Batch operacijski sustav 3
3.3.2. Time-sharing operacijski sustav 3
3.3.3. Raspodijeljeni operacijski sustav 4
3.3.4 Mrežni operacijski sustav 4
3.3.5. Operacijski sustav u stvarnom vremenu 4
4. Povijesni razvoj operacijskih sustava 5
4.1. Prvi operacijski sustavi 6
4.2. Daljnji razvoj operacijskih sustava 7
4.3. Kasniji i moderni operacijski sustavi 8
4.3.1. Unics (UNIX) 8
4.3.2. RSTS 9
4.3.3. Alto OS 10
4.3.4. QDOS, 86-DOS, PC-DOS i MS-DOS 10
4.3.5. GNU 11
4.3.6. Mac OS 12
4.3.7. Microsoft Windows 14
4.4. Kratka povijest Linuxa 26
5. Zadaci operacijskog sustava 28
6. Von Neumannov model računala 29
6.1. Funkcijski model računala 29
6.2. Sabirnička građa računala 30
6.3. Računalni proces 31
7. Ulazno-izlazne naprave 33
7.1. Prijenos pojedinačnih znakova 35
7.2. Prijenos blokova znakova neposrednim pristupom spremniku 38
7.3. Sabirnički sustav stvarnih računala 40
8. Programi, procesi i dretve 41
8.1. Općenito o procesima 41
8.2. Model višedretvenosti 42

8.3. Sustav Dretvi 45
8.4. Međusobno isključivanje dretvi 46
8.5. Cikličke Dretve 47
9. Jezgra operacijskog sustava 48
9.1. Jednostavni model jezgre 49
9.1.1. Struktura podataka jednostavnog modela jezgre 51
9.2 Jezgrine Funkcije 53
9.2.1. Funkcije za ulaz i izlaz iz jezgre 53
9.2.2. Funkcije za binarni semafor 54
9.2.3. Funkcije za opći semafor 55
9.2.4. Funkcije za ostvarivanje kašnjenja 56
9.2.5. Funkcije za ulazne i izlazne operacije 57
10. Datotečni sustav 58
10.1. Uloga datoteka u računalnim sustavima 58
10.2. Struktura datoteka 59
10.2.1. Binarna datoteka 59
10.2.2. Nestrukturirana datoteka 59
10.3. Smještanje datoteka na disk 60
11. Anketa 61
12. Zaključak 64
12. Literatura 65
13. Popis Slika 67
14. Popis Tablica 68

1. Uvod

U današnje vrijeme, stalnim ubrzanim razvitkom informacijske tehnologije i programske potpore računalnog sustava, računala se danas koriste u skoro svim područjima ljudskog djelovanja. Računala su postala jednostavna za korištenje i dostupna su gotovo svima. Napretkom tehnologije život bez računalnih uređaja je nezamisliv, dok je posao teško ostvariv.

Unutar računala postoji softver koji je ujedno i najkompliciranija stavka računala i bez koje rad računala nije moguć niti ostvariv. Softver računalnog sustava se dijeli na sustavni i aplikacijski softver. Sustavni softver se dijeli na: operacijski sustav, uslužni softver i jezike prevoditelje. Upravo je operacijski sustav na računalima tema ovog diplomskog rada.

„Za operacijski sustav može se reći kako je to skup programa koji omogućuje upravljanje računalnim hardverom i pruža podršku za korištenje programa i omogućava komunikaciju korisnika sa računalom.“ [2]. Glavni zadatak operacijskog sustava je taj da optimizira korištenje hardvera. Operacijski sustavi dizajnirani su na način da podržavaju apsolutno sve, od kompleksnih videoigara do poslovnih aplikacija i ostalih aplikacija za najzahtjevnije korisnike.

Cilj ovog diplomskog rada je detaljnije opisati rad operacijskog sustava, reći će se ukratko nešto o povijesti operacijskih sustava i kako je sve krenulo. Zatim će se detaljnije opisati rad programa, procesa, dretvi, sabirnica, jezgre operacijskog sustava, datotečnog sustava i ostalih bitnih stvari koje su ključne za pravilan rad operacijskog sustava. U diplomskom radu se neće pričati o sigurnosti operacijskih sustava.

2. Metode i tehnike rada

Razradom navedene teme diplomskog rada koristile su se razne metode proučavanja i istraživanja. Stručna literatura, stručne knjige i znanstveni članci su bili najčešće korišteni u ovom diplomskom radu kako bi se bolje razradila tema. Pretraživan je internet gdje se je došlo do mnogih podataka koji su bili bitni za pisanje ovog diplomskog rada.

3. Podjela operacijskih sustava

U ovom poglavlju navesti će se i opisati podjele operacijskih sustava s obzirom na tip koda i s obzirom na izvođenje dretvi. Nakon što su opisane podjele reći će se nešto o tipovima operacijskih sustava [12].

3.1. Podjela operacijskih sustava s obzirom na tip koda

Kao što je već ranije navedeno operacijske sustave s obzirom na tip koda možemo podijeliti na otvorene i zatvorene. Razlika između operacijskih sustava otvorenog i zatvorenog programskog koda je ta da kod otvorenih svaka osoba može vidjeti i mijenjati kod dok se to kod zatvorenih operacijskih sustava ne može jer je njihov programski kod zaštićen autorskim pravima. Kod otvorenog OS-a korisnici mogu i ispravljati greške i na taj način poboljšati operacijski sustav. Operacijski sustavi otvorenog koda su: BSD i Linux, a operacijski sustavi zatvorenog koda su: Windows, Digital Research i UNIX [12].

3.2. Podjela operacijskih sustava s obzirom na izvođenje dretvi

Što se tiče operacijskih sustava s obzirom na izvođenje dretvi oni se dijele na:

- Jednodretvene – takvi operacijski sustavi mogu izvoditi samo jednu dretvu instrukcija i u današnje vrijeme se više ne koriste.
- Višedretvene. – ovi operacijski sustavi mogu istovremeno izvoditi više dretvi i u današnje vrijeme se samo oni koriste.

3.3. Tipovi operacijskih sustava

U ovom poglavlju reći će se nešto više o tome koji tipovi operacijskih sustava postoje.

3.3.1. Batch operacijski sustav

Batchov operacijski sustav je bio jedan od prvih operacijskih sustava. On radi na principu sustava programa koji se izvode u grupama. Sustav grupira sve programe koji se trebaju izvesti i nakon toga ih izvršava. Ovo je jako star tip operacijskih sustava i danas ih ne susrećemo ili iznimno rijetko može susresti ovakav tip operacijskog sustava [12].

3.3.2. Time-sharing operacijski sustav

Radi na principu da omogućava korisnicima sa različitih lokacija pristup jednom računalu. S obzirom da više korisnicima treba pristupiti određenom računalu, glavni cilj Time-sharing operacijskih sustava je smanjiti vrijeme odgovora korisniku što se može postići na

način da se svakom korisniku dodijeli *„fiksni, mali interval u kojem se procesor bavi njegovim zahtjevom“* [12].

3.3.3. Raspodijeljeni operacijski sustav

Ovaj tip operacijskog sustava upravlja s više računala koja imaju svoje vlastito sklopovlje, spojena nekom vrstom komunikacijske mreže, ali korisniku izgleda kao jedno računalo [12].

3.3.4 Mrežni operacijski sustav

Mrežni operacijski sustav (engl. *Network operating System*) sastoji se od središnjeg servera odnosno računala koje ima operacijski sustav na koji su spojena različita računala u svrhu podjele podataka. Sva računala su nezavisna i operacijski sustav ima pristup samo računalnom sklopovlju računala na kojem se nalazi i ograničen pristup podacima na računalima spojenim u mrežu [12].

3.3.5. Operacijski sustav u stvarnom vremenu

Operacijski sustav u stvarnom vremenu je tip operacijskog sustava konstruiran na način da obrađuje podatke čim su isti uneseni. Neki podaci ne mogu biti obrađeni odmap pa se onda prvo obrade najprioritetniji, na taj način vrijeme odgovora računala korisniku maksimalno je smanjeno. *„Jako je bitno da su ovi sustavi predvidivog ponašanja, omogućeni za rad u svim uvjetima i jednostavni“* [12].

4. Povijesni razvoj operacijskih sustava

1940-tih godina prošlog stoljeća pojavila su se prva računala. Tadašnja računala su samo mogla pozivati potprograme. Mogla su adresirati 4.096 riječi i bilo je teško uopće razmišljati o operacijskim sustavima na takvim računalima. 1950-tih godina pojavila su se računala koja su pohranjivala podatke na magnetske trake te su imala indeks-registre i instrukcije za prijelaz u potprograme. Do sredine 1950-tih godina počela su se pojavljivati računala s više čitača magnetskih traka. Kako je vrijeme prolazilo postalo je uobičajeno da se na jednoj od tih manetskih traka koja su karakteristična za to računalo nalazi program za učitavanje programa s ostalih magnetskih traka. To nazivamo prvim počecima operacijskih sustava. Oni su u samo nekoliko riječi pamtili gdje se na traci nalazi interpreter komandne linije za inicijalizaciju i pokretanje ostalih programa. Na kraju svog rada svaki je program trebao vratiti komandu operacijskom sustavu, inače bi se računalo moralo ponovno pokrenuti. Ako je neki program slučajno promijenio podatke učitavača ostalih programa rezultat je bio potpuno nepredvidiv pa su takva računala bila softverski vrlo krhka, unatoč tome ti su sustavi svejedno bili sposobni pozivati biblioteke te su podržavali assemblere, prevoditelje i poveznike [1].

4.1. Prvi operacijski sustavi

GM-NAA I/O - 1956. godine nastao je operacijski sustav koji se zvao GM-NAA I/O, to je bio prvi operacijski sustav za IBM-704 računalo. Napravljen je zajedničkom suradnjom General Motors i Sjeverno-Američke avijacije. Glavna funkcija mu je bila da nakon završetka jednog programa odmah pokreće sljedeći program.

UMES (UNIVERSITY OF MICHIGAN EXECUTIVE SYSTEM) - je razvijen na Sveučilištu u Michiganu 1958. godine. Radio je na IBM-701 računalu i bio je optimiziran za male programe s kojima su radili tadašnji studenti.

SHARE - 1959. godine izbačen je operacijski sustav pod nazivom SHARE, nastao je kao unapređenje GM-NAA I/O-a. SHARE je također radio na IBM-704 računalima kao i GM-NAA I/O, ali je kasnije prenesen na IBM-709 računala. SHARE je uveo nove metode za upravljanje privremenom memorijom te ulazno-izlaznim uređajima. Kasnije ga je IBM prebacio i na 7090 i 7094 računala pod imenom IBSYS.

ATLAS **-** se zvao operacijski sustav i računalo nazvano koje je razvijeno 1962. godine na sveučilištu u Manchesteru u Engleskoj. To računalo je imalo poseban operacijski sustav koji se zvao Atlas Supervisor i on je omogućavao da se više programa izvodi istovremeno.

COMPATIBLE TIME-SHARING SYSTEM (CTSS) - je prvi put demonstriran 1961. godine na MIT-u i upotrebljavan je sve do 1973. godine. Riječ kompatibilan (engl. *compatible*) u nazivu odnosi se na to da je bio kompatibilan s uobičajenim operacijskim sustavom za IBM -7094 računala. CTSS je utjecao na razvoj mnogih kasnijih operacijskih sustava od kojih se neki poput UNIX-a koriste i danas. Najvažnije karakteristike koje je CTSS uveo bile su:

- mogućnost izvršavanja više programa paralelno
- mogućnost korisnika da jedni drugima ostavljaju poruke i naredbu RUNCOM (koju je napravio Louis Pouzin) koja je izvršavala niz naredbi iz datoteke te je podržavala substituciju parametara za te naredbe.

MASTER CONTROL PROGRAM (MCP) – je bio vlasnički operacijski sustav za Burroughs Large Systems koji je prvi put postao javno dostupan 1961. godine te se koristi i danas. Iako je korporacija Burroughs osnovana još davne 1886. godine mnogo prije dolaska električnih računala, ta korporacija se relativno kasno uključila u proizvodnju električnih računala. Tako je MCP uveo mnoge novosti u razvoj operacijskih sustava i računala općenito. Neki od tih

noviteta tek su se nedavno počeli masovno implementirati u današnje operacijske sustave. Izvorno je napisan u ESPOL-u (Executive Systems Programming Language) ali je 1970. godine prebačen i napisan u NEWP-u (New Executive Programming Language) koji je bio bolje strukturiran i sigurniji od ESPOL-a. MCP je prvi operacijski sustav koji je podržavao višestruke procesore te ujedno posjeduje i prvu komercijalnu implementaciju virtualne memorije. MCP je prvi operacijski sustav pisan isključivo u višem programskom jeziku. Hardver na kojem se u početku pokretao razvijan je u skladu s potrebama softvera, tako se i danas radi. Iako je izvorni kod MCP-a vlasnički, korisnici imaju uvid u njega te mu mogu pridonijeti. Tako su mnoga proširenja i zakrpe koje su napisali korisnici dospjele čak do osnovnog koda samog operacijskog sustava [1].

4.2. Daljnji razvoj operacijskih sustava

Nadalje razvijaju se idući operacijski sustavi koji će biti samo nabrojani u nastavku i njihova godina izlaska prve verzije u javnost:

- 1962. godine – GCOS (General Comprehensive Operating System).
- 1964. godine – EXEC 8 i TOPS-10.
- 1965. godine – OS/360.
- 1964. godine – Berkeley Timesharing System.
- 1967. godine – ITS (Incompatible Timesharing System).
- 1968. godine – CP/CMS.
- 1969. godine – MULTICS.
- 1972. godine – VM/370 i z/VM [1].

4.3. Kasniji i moderni operacijski sustavi

4.3.1. Unics (UNIX)

Unics ili UNIX je prvi put posato javno dostupan 1969. godine. On je Izvorno razvijan u korporacijama AT&T-u i Bell Labs-u. Kasnije kada je počeo podržavati simultan rad dvaju korisnika, naziv mu je promijenjen u UNIX. Danas naziv "UNIX" koristi se za svaki operacijski sustav koji sadrži UNIX-ove standarde i kodove. Današnji UNIX sustavi se razvijaju i dalje u AT&T-u, te u raznim komercijalnim i nekomercijalnim tvrtkama. Naziv "UNIX" dodjeljuje se samo operacijskim sustavima koji se u potpunosti kvalificiraju po „Single UNIX Specificationu“ kojeg definira „The Open Group“, koji su ujedno i nositelji UNIX-ove zaštitne oznake. Mnogi operacijski sustavi se zato nazivaju UNIX-olikim (engl. UNIX-like), iako se „The Open Group“ ne slaže s tim izrazom. Utjecaj UNIX-a u akademskim krugovima je doveo do masovnog prihvaćanja UNIX-a, posebice BSD (Berkley Software Distribution) varijante u komercijalim operacijskim sustavima kao što su: Mac OS X, Solaris, HP-UX i AIX. Osim certificiranih UNIX operacijskih sustava, danas postoje i mnogi sustavi kao što su razne varijante BSD-a i nebrojene distribucije GNU/Linux-a. Izraz "tradicionalni Unix" se koristi za operacijske sustave koji u potpunosti nose sve karakteristike originalnog UNIXA-a, kao što su Version 7 UNIX ili UNIX System V. UNIX operacijski sustavi se koriste i za poslužitelje i za korisnička računala. UNIX okruženje i klijentsko poslužiteljska struktura bili su veoma važni za razvoj Interneta i reorganizaciju računarstva orijentiranog prema mrežama, a ne zasebnim računalima. UNIX je dizajniran da bi bio prenosiv na druge platforme, te da bi više korisnika moglo u dijeljenom vremenu izvoditi više programa istovremeno. U UNIX-u se operacijski sustav sastoji od raznih alatnih programa i glavnog upravljačkog programa Kernela. Kernel se brine o poslovima niže razine kao što su pokretanje i zaustavljanje programa, pristup datotečnom sustavu i pristup sklopovlju. Kako bi kernel mogao uspješno obavljati svoje zadaće, dane su mu posebne ovlasti nad sklopovljem te je nastala podjela za izvršavanje operacija na operacije koje se izvode u takozvanom korisničkom prostoru (engl. *user-space*) i operacije koje se izvode u kernel-prostoru (engl. *kernel-space*) [1].

4.3.2. RSTS

RSTS je bio višekorisnički operacijski sustav s dijeljenjem vremena razvijen u Digital Equipement Corporation (DEC), koji je sada dio Hewlett Packarda. Radio je na PDP-11 seriji 16-bitnih miniračunala. Prva verzija je napravljena 1970. godine, a zadnja verzija je izašla 1992. godine. Prve verzije su nosile naziv RSTS-11, dok su zadnje verzije nosile naziv RSTS/E. Njegov Kernel je pisan u Assembleru, a za svoj rad je zahtijevao samo 64KB memorije. Za rad mu je bio potreban i memorijski prostor za ulazno-izlazni izravni pristup memoriji. Zahvaljujući svojoj štedljivosti na memoriji, mogao je imati do 16 terminala s najviše 17 poslova. Maksimalna veličina pojedinih programa je bila 16KB. 1973. godine su dodali upravljanje memorijom za tada nova DEC-ova PDP-11/40 i PDP-11/45 miniračunala. Do 1975. godine je RSTS mogao koristiti do 2MB memorije na PDP-11/70 računalima pa je mogao podržavati do 63 posla u danom trenutku. Pojedini poslovi su tada mogli kontrolirati do čak 128 terminala. Imao je izvrsnu komunikaciju među procesima. Iako su prve verzije trebale DOS-11 operacijski sustav za instalaciju, od 1977. godine se instalacija odvijala pod RT-11 operacijskim sustavom. Od 1978. godine je podržavao i dijagnozu s udaljenog računala. Od 1981. godine se moglo u programima odvojiti memoriju za programski strojni kod i podatke, pa su programi mogli koristiti do 64KB memorije za programski kod i dodatnih 64KB memorije za podatke. 1983. godine nadograđen je datotečni sustav, koji je mogao podržati do 1.700 korisničkih računa. Za razliku od UNIX-a koji ima samo jedan „root“ privilegirani korisnički račun RSTS je imao cijelu korisničku grupu privilegiranih računa. DEC je 1994. godine prodao svoj softver za PDP-11 tvrtki Mentec, koja je kasnije omogućila hobistima da besplatno dobiju licencu za nekomercijalnu upotrebu. RSTS se danas može pokretati pod simuliranim hardverom na osobnim računalima, a prema želji korisnika i brže nego se taj sustav originalno pokretao na pravom hardveru [1].

4.3.3. Alto OS

Xerox Alto je bilo rano osobno računalo, ujedno i prvo koje se nazivalo stolnim računalom (engl. *desktop*). Sadržavalo je grafičko korisničko sučelje. Računalo nije bilo komercijalan proizvod, imalo je velik utjecaj na daljnji razvoj osobnih računala, a posebice Macintosha i Sunovih radnih stanica. Njegov operacijski sustav (Alto OS) je bio prvi operacijski sustav koji je koristio miš (engl. *mouse*) za navigaciju kroz grafičko korisničko sučelje, što je danas apsolutni standard za sva osobna računala [1].

4.3.4. QDOS, 86-DOS, PC-DOS i MS-DOS

Izvorno je bio poznat pod imenom QDOS (engl. *Quick and Dirty Operating System*), ali nedugo nakon toga preimenovan je u 86-DOS kada ga je kupila tvrtka Seattle Computer Products. Njegova komandna struktura i sučelje s aplikacijskim programima su imitirali onu CP/M-a, pa je bilo lako prenijeti postojeće programe s CP/M-a na 86-DOS. Napravljen je zato jer SCP-ova 8086 bazirana računala nisu imala operacijski sustav, već su mogla raditi samo sa Microsoft-ovim BASIC-86. Kompanija Seattle Computer Products je željela ponuditi verziju CP/M-a za 8086, ali nisu mogli čekati da isti izađe. Stoga su u SCP-u odlučili napraviti sustav kompatibilan sa CP/M-om. Promijenili su neke značajke koje im se nisu sviđale kod CP/M-a. Tako su CP/M-ovo privremeno pohranjivanje podataka koji se pišu na disk u memoriju radi bržeg rada zamijenili sa sporijom ali sigurnijom tehnikom izravnog zapisivanja podataka na disk kako korisnici ne bi zaboravili ručno sinkronizirati podatke na diskovima. Datotečni sustav nije bio isti kao u CP/M-u, već se koristio FAT datotečni sustav zbog bolje kompatibilnosti s već prodanim SCP-ovim sustavima. Pred kraj 1980. godine IBM je razvijao IBM Personal Computer. U tadašnje vrijeme najpopularniji operacijski sustav je bio CP/M. IBM ga je želio za svoja računala kako bi što bolje uspio na tržištu. U isto to vrijeme Microsoft je kupio 86-DOS i prilagodio ga 8088 procesoru. Budući da je izgledao poput CP/M-a i da je bio kompatibilan s njime IBM se svejedno odlučio za njega. IBM PC verzija 86-DOS-a je nazvana PC-DOS. 1982. godine IBM je zatražio od Microsofta da naprave DOS kompatibilan s tvrdim diskovima što je Microsoft i učinio. To je bio PC-DOS 2.0 koji je bio gotovo u potpunosti prerađen te je vrlo malo originalnog koda QDOS-a ostalo u njemu. No jedan program je ostao još od 86-DOS-a do danas (EDLIN). Radi se o primitivnom linijskom tekst editoru koji dolazi ugrađen u emuliranom DOS okruženju sve do Windows 7. IBM i Microsoft su surađivali na DOS-u sve do 1993. godine. Do tada nije bilo znatne razlike između PC-DOS-a i MS-DOS-a, osim što je PC-DOS bio prodavan s IBM-ovim računalima, a MS-DOS je bio za otvoreno tržište [1].

4.3.5. GNU

Naziv GNU je kratica za GNU nije UNIX (engl. *GNU's not UNIX*). Taj naziv je izabran zato što je operacijski sustav kao UNIX, ali ne sadrži originalni UNIX-ov kod i u potpunosti je slobodan softver. Osnovni sustav sastojao se od:

- GNU Compiler Collection (GCC)
- GNU Binary Utility
- Bash školjke
- GNU C library
- GNU Core Utility

Iako su gotovo sve komponente GNU-a dovršene prije više od petnaest godina. GNU Hurd nije dovršen iako je relativno stabilan. Mnoge komponente GNU-a su prenesene i na razne druge platforme kao što su BSD, Solaris i Mac OS X. Operacijski sustav GNU u kombinaciji s Linux kernelom naziva se GNU/Linux ili svima poznatije samo Linux. GNU-ove licence GNU General Public License (GPL), GNU Lesser General Public License (LGPL) i GNU su originalno namjenjene GNU-u, ali se danas najviše koriste za razne projekte koji nisu vezani za GNU. Cilj projekta je bio je napraviti potpuno slobodan operacijski sustav kako bi korisnici bili slobodni kao 1960-tih i 1970-tih da mogu proučavati izvorni kod softvera i da ga mogu slobodno mijenjati. Pošto je tada UNIX bio veoma popularan i budući da se pokazao vrlo stabilnim i pouzdanim i da je bio modularan moglo ga se je reimplementirati dio po dio. Richard Stallman, koji je pokrenuo GNU projekt, odlučio ga je napraviti nalik na UNIXU. Kod za GNU su pisale razne osobe različitih struka od dobrovoljaca do plaćenih programera koje su financirale razne neprofitabilne organizacije. Dio već postojećeg slobodnog softvera iskorišten je za GNU, kao što je X Window System. Originalno je zamišljen da bi bio kompatibilan s UNIX-om uz prijedlog poboljšanja gdje je to bilo potrebno [1].

4.3.6. Mac OS

1984. godine Apple je razvio Macintosh osobno računalo koje je dolazilo s operacijskim sustavom koji je bio poznat tada kao System Software. Kasnije je System Software operacijski sustav preimenovan u današnji Mac OS. Mac OS se nalazi instaliran na gotovo svakom novom Macintosh računalu te ga je moguće kupiti i odvojeno. Prvotni Mac OS je bio djelomično zasnovan na Lisa operacijskom sustavu, koji je koristio Apple-ovo računalo isto naziva Lisa. Lisa operacijski sustav je bio napredan za svoje vrijeme i imao je mnoge osobine koje su rijetki operacijski sustavi tog vremena imali. Sastojao se od grafičkog sučelja, suradničke višezadaćnosti i virtualne memorije. Jedan od nedostataka operacijskog sustava Lisa bila je velika količina naprednih funkcija koja je mnogim korisnicima otežavalo korištenje sustavom te se sustav činio sporim. Macintosh je bio zamišljen kao jednostavno računalo za uporabu i jeftino računalo za prosječnog korisnika. U Apple-u su izmislili izbornike, skočne izbornike i koncept „drag and drop“. Prva verzija Mac OS-a, zvana jednostavno System i bila je lako prepoznata od ostalih operacijskih sustava jer nije imao komandnu liniju. Bio je to jedan od prvih operacijskih sustava koji su koristili u potpunosti grafičko korisničko sučelje. Osim samog kernela tu je još bio i Finder (aplikacija koja je radila s datotekama sustava i prikazivala radnu površinu). Datoteke samog Kernela i Findera su se nalazile u mapi nazvanoj „System Folder“, koja je još sadržavala nekoliko drugih datoteka za pristupanje sustavu kao što je upravljački program za pisač. Prve četiri verzije Mac OS-a mogle su raditi samo s jednom aplikacijom u danom trenutku. Prvotne verzije Mac OS-a su imale ravni datotečni sustav sa samo jednom razinom direktorija koji se zvao Macintosh File System (MFS). U System 2.1 koji je imao Finder 5.0 uveden je hijerarhijski datotečni sustav Hierarchial File System (HFS), on je imao prave direktorije i poddirektorije. System 4.0 je imao boje, veći zaslon i noviji Motorola 68020 procesor. U System 5 je dodana inačica MultiFinder (proširenje koje je dopuštalo da pokreće više aplikacija od jednom). Sustav je koristio suradničku višezadaćnost, odnosno pozadinski programi su dobivali procesorsko vrijeme samo kada bi to aktivna aplikacija odlučila. System 6 je radio na 68030 procesoru. Kasnije verzije System-a 6 su uključivale podršku za prve specifične značajke prijenosnog računala kada je Apple napravio Macintosh Portable. System 7 je uveo velike promjene u sustav, podržavao je 68040 procesor. Korisničko sučelje je dobrim djelom prerađeno te su dodane mnoge nove mogućnosti. Također je dodano 32-bitno adresiranje, koje je doduše stvaralo probleme korisnicima koji su koristili starije 68000 procesore i oni su imali 24-bitnu adresnu sabirnicu. Kasnije su dodani AppleScript, TrueType fontovi, Apple menu, smeće (engl. *Trash)* je prerađeno tako da podaci u njemu ostaju sve dok ih korisnik ne bi namjerno obrisao. „System 7.1 je bio prvi u seriji koji se naplaćivao. Nakon verzije 7.1.2 je dodana podrška za PowerPC temeljene Mac-ove, iako je na njima sustav radio poprilično nestabilno. Od verzije 7.6 ime mu je promijenjeno u Mac OS. Mac OS 7.6 je

zahtijevao 68030 procesor pa je time prekinuta podrška za mnoge starije Mac-ove. U Mac OS-u 8 je uključen višedretveni Finder, što je omogućilo bolju višezadaćnost. Također je dodana mogućnost mijenjanja boja korisničkog sučelja. U verziji 8.1 je datotečni sustav nadograđen na HFS Plus, koji je riješio mnoga ograničenja starog HFS-a. HFS Plus se koristi i u Mac OS X-u. Mac OS 9 je dodao podršku za bežične mreže i neke prijenosne tehnologije kako bi programerima olakšali prijelaz na Mac OS X. Time je omogućeno da neke aplikacije s Mac OS-a 9 izravno mogu raditi na Mac OS X-u. Mac OS 9 se također može pokretati kao aplikacija pod Mac OS X-om. Za razliku od prethodnih verzija, Mac OS X se zasniva na Unix tehnologiji razvijanoj u NeXT-u dok Apple nije kupio NeXT. Dodane su podrške za razne programske jezike kao što su C, C++, Objective-C, Java i Python. Korisničko sučelje je prerađeno i nazvano Aqua [1].„



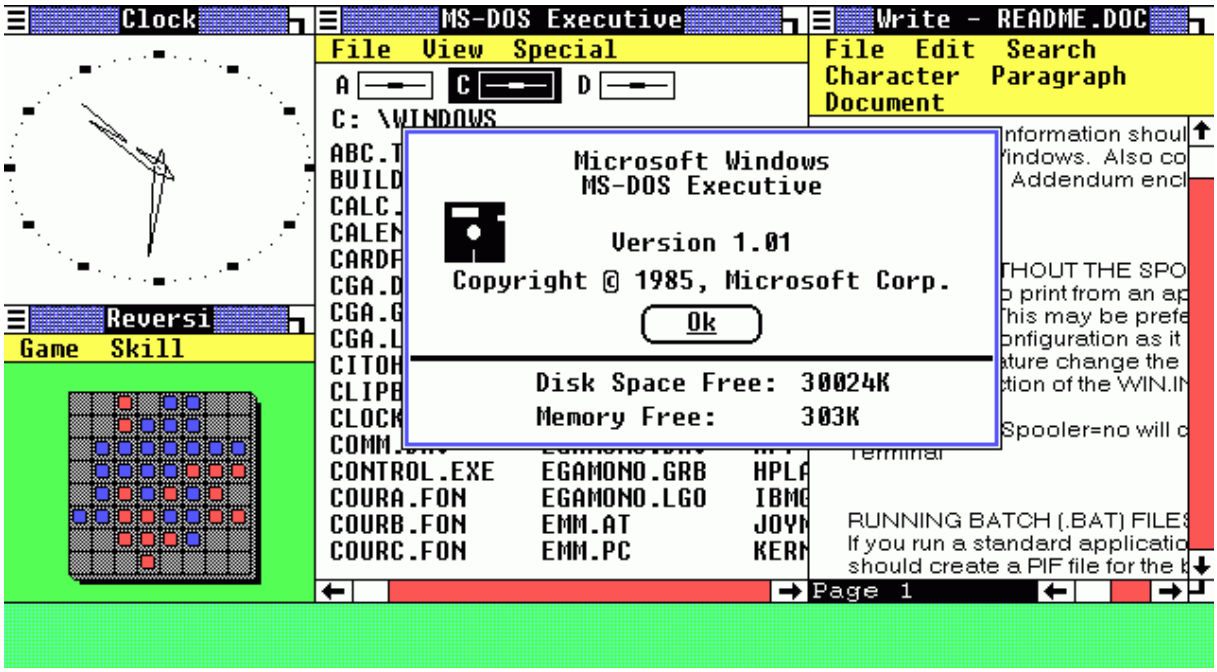
Slika 1: Prikaz modernog Mac OS-a

(*Izvor: https://www.computerworld.com/article/3285791/10-macos-10-14-mojave-secrets.html)*

4.3.7. Microsoft Windows

Microsoft je 1983. godine započeo razvoj Windowsa, drugim riječima grafičkog korisničkog sučelja za MS-DOS. Windowsi su se od tada razvili u moderan operacijski sustav koji je danas najpopularniji i najkorišteniji na računalima. Iako su starije verzije radile i na Alpha-i, MIPS-u, PowerPC-u i još nekim arhitekturama, danas rade samo na x86-x64 i Itanium procesorima [1].

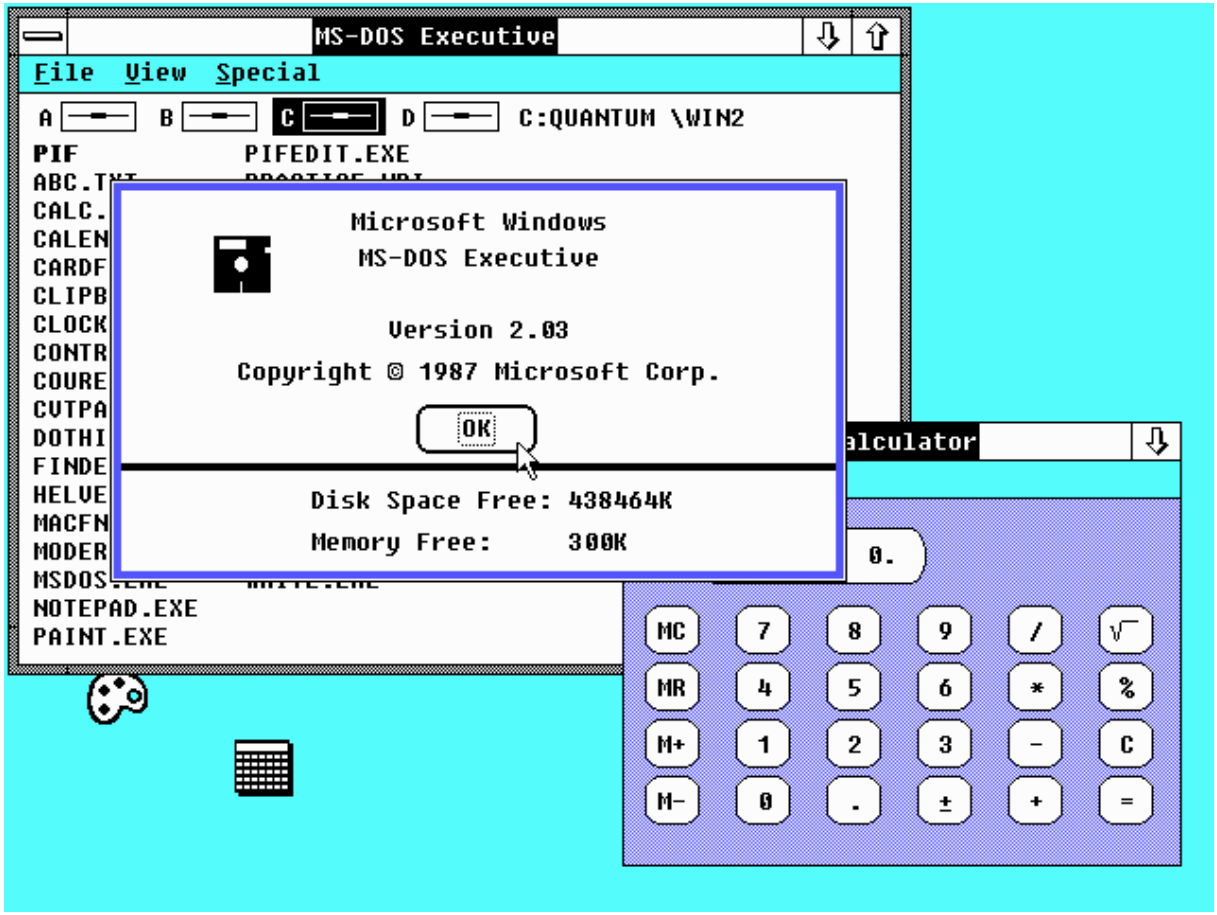
Prva verzija Windowsa (Windows 1.0) je izašla 1985. godine, ali nije stekla veliku popularnost. To nije bio operacijski sustav, već samo operacijsko okruženje koje je proširivalo MS-DOS. Prvi Windowsi su sa sobom uključivali programe: Windows Paint, Windows Write, Kalendar, Notepad, Sat i još nekoliko programčića. Prozori se nisu mogli preklapati, već su samo mogli biti jedan pored drugog. Ispod na slici je prikaz prve verzije Windowsa [1].



Slika 2: Prva verzija Windowsa (Windows 1.0)

(Izvor: https://povijest.hr/nadanasnjidan/kako-je-to-nekad-izgledalo-microsoft-windows-1-0-1985/)

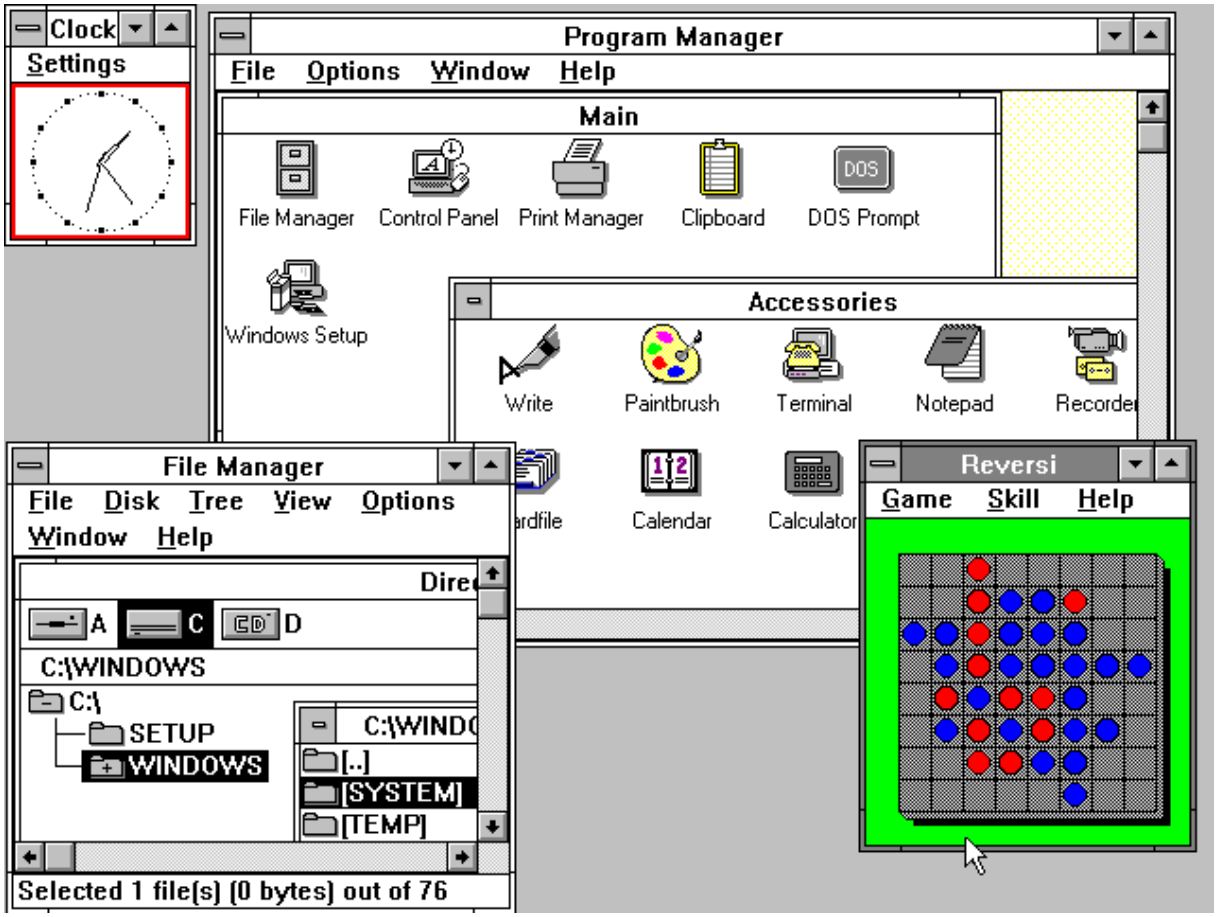
Druga verzija Windowsa (Windows 2.0) je izašla 1987. godine i bila je nešto popularnija od prethodne verzije. Ova verzija je uključivala „run-time“ verziju, koja se pokretala samo za vrijeme pokretanja jedne aplikacije nakon čega su se Windowsi gasili i korisnik se vratio u MS-DOS komandnu liniju. Ovo se posebice koristilo s Excel-om i Word-om za Windowse. Windows 2.0x su se mogli pokretati pod drugim višezadaćnicima, poput DESQview-a. Windowsi 2.1 su došli u dvije varijante: Windows/286 2.1 i Windows/386 2.1. te su bili prvi koji su mogli koristiti više od 1MB memorije [1].



Slika 3: Izgled druge verzije Windowsa (Windows 2.0)

(Izvor: https://www.cs.umd.edu/hcil/muiseum/systems/win2src.html)

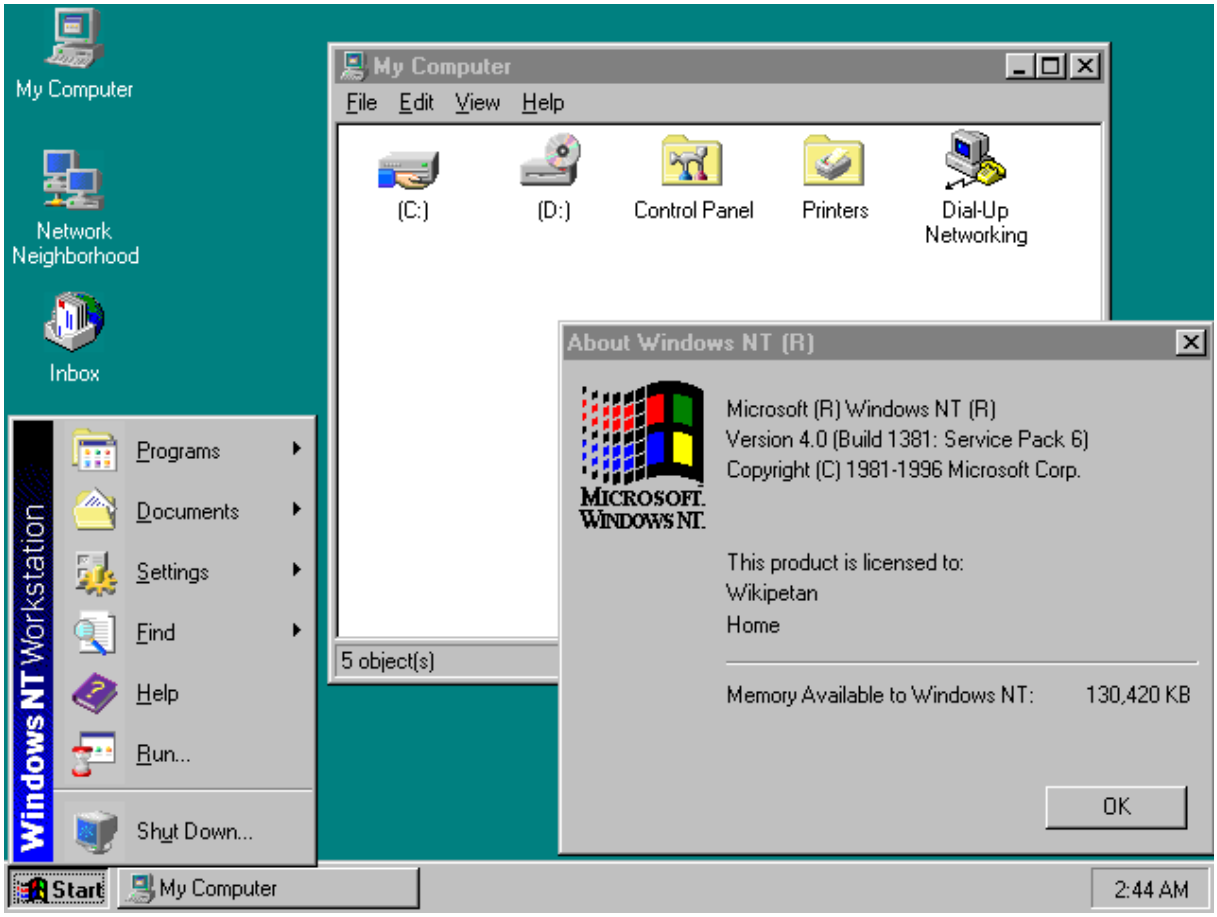
1990. godine nastao je Windows 3.0, oni su uključivali virtualnu memoriju što je znatno unaprijedilo višezadaćnost starih MS-DOS aplikacija. Windowsi 3.0 su također imali mnogo bolje sučelje zahvaljujući razvoju grafičkih kartica za računala. Nekoliko mjeseci kasnije Microsoft je objavio Windows 3.0 sa Multimedia Extensions 1.0, koji su mogli iskoristiti CD-ROM pogon i zvučne kartice. Zahvaljujući tome Windowsi su dosegli veliku popularnost i bili su podržani sve do kraja 2001. godine. Kasnijim unaprijeđenjem na Windows 3.1 oni su dodali podršku za TrueType pisma te su proširili podršku za multimediju. Radili su samo na 80286 ili boljem procesoru, imali su podršku za 8086 i 8088. Kasnije je Microsoft razvio i Windows 3.11, koji su uključivali mnoge zakrpe u odnosu na Windows 3.1. Uskoro su izašli i Windows for Workgroups koji su imali poboljšanu podršku za rad u mreži [1].



Slika 4: Microsoft Windows 3.0

(Izvor: https://timesofindia.indiatimes.com/how-microsoft-windows-has-changed-over-the-last-30-years/Windows-3-0-1990/photostory/45993297.cms)

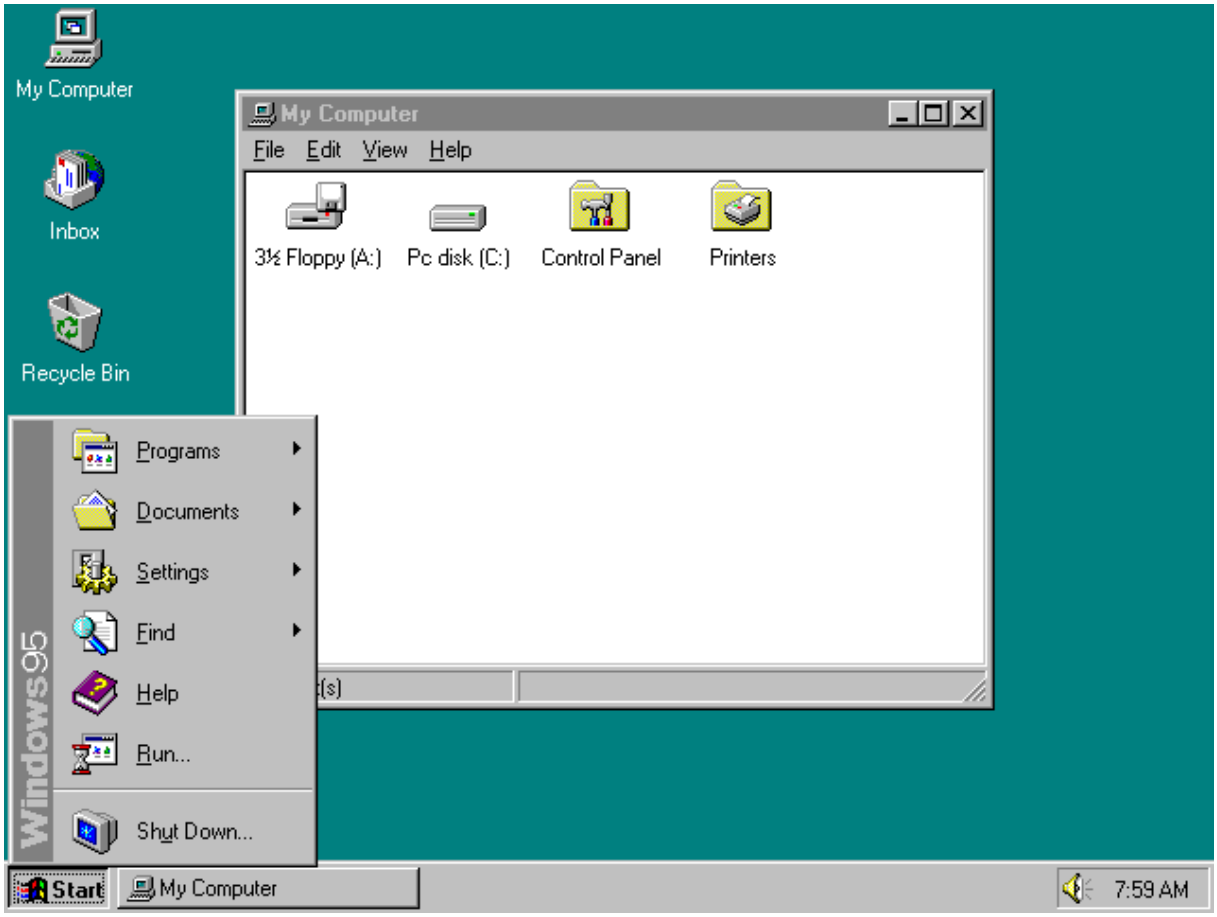
Windows NT nose isto ime Windows, ali se radi o potpuno drugačijem operacijskom sustavu. Windowsi NT su imali mnogo poteškoća s upravljačkim programima, jer je mnogim programerima bilo izrazito teško pisati kod za Windows NT. Bili su veoma zahtjevni pa su bili isplativi samo na velikim računalima. Imali su mnogo bolju podršku za umreženi rad od Windows 3.11 i Windows 95 koji su se paralelno razvijali. Windows NT je imao svoj datotečni sustav (NTFS) koji se koristi i danas [1].



Slika 5: Windows NT

(Izvor: https://en.wikipedia.org/wiki/Windows_NT_4.0)

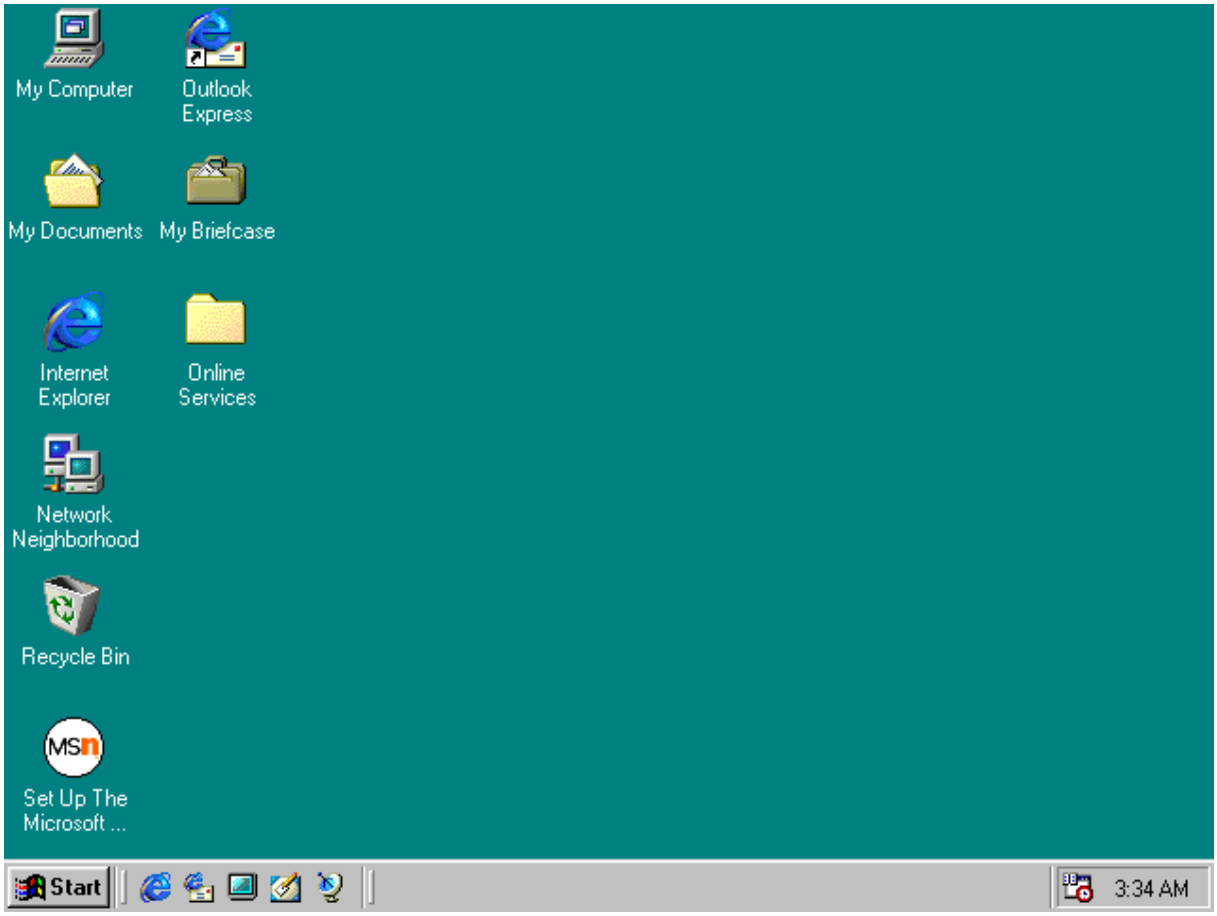
Windows 95 su imali naizmjeničnu 32-bitnu višezadaćnost i 32-bitno programsko sučelje iz Windowsa NT-a, ali neki dijelovi operacijskog sustava su ostali 16-bitni. Windows 95 su koristili DOS na kratko vrijeme prilikom podizanja sustava. Imali su virtualnu memoriju te su mogli koristiti do 2GB memorije za sustav i još 2GB za aplikacije, ali u praksi nije bilo moguće koristiti više od 512MB fizičke memorije bez razvijenije konfiguracije. Isto tako je ostalo kroz Windows 98 i Windows ME. Windows 95 su također imali drugačije i novije korisničko sučelje. Kasnijih godina su razvijeni i Windows NT 4.0, koji su imali isto korisničko sučelje, ali su sadržavali i NT kernel [1].



Slika 6: Prikaz Windowsa 95

(Izvor: https://en.wikipedia.org/wiki/Windows_95)

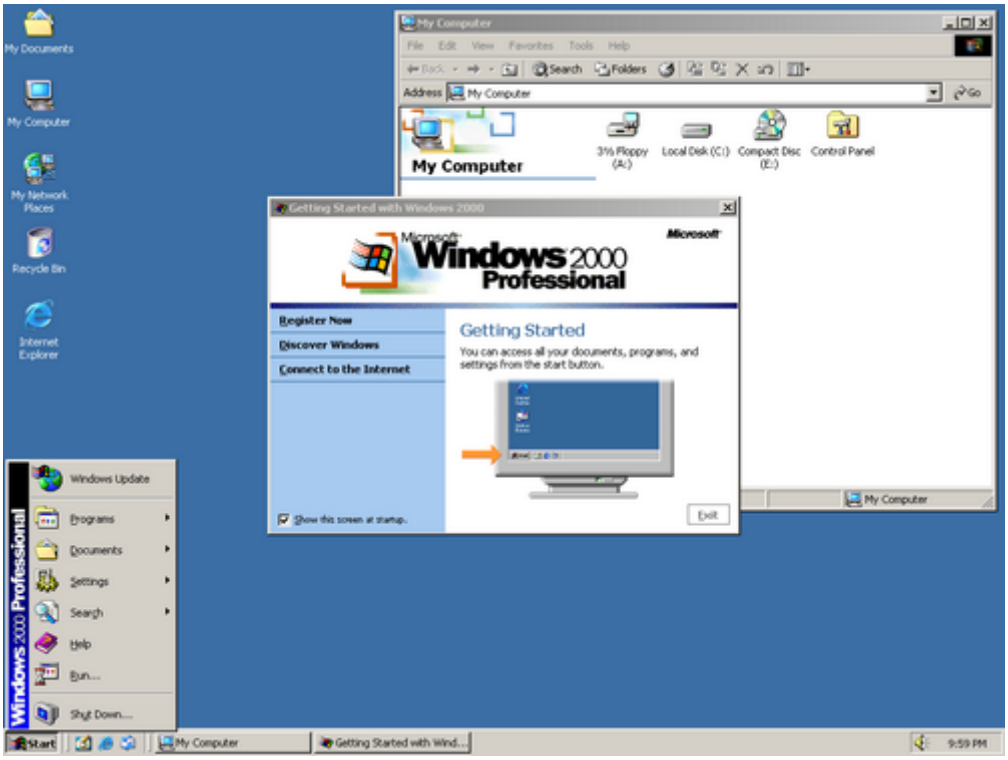
Windows 98 kao što im samo ime govori nastali su 1998. godine te su imali mnogo bolju podršku za hardver i potpunu podršku za FAT32 datotečne sustave. Datotečni sustav FAT32 je omogućio upotrebu tvrdih diskova s kapacitetom koji su veći od 2GB. Integrirali su Internet Explorer u korisničko sučelje. Kasnije je izašla verzija Windows 98 SE, koji su omogućili dijeljenje Internet veze jednog računala s ostalim računalima na lokalnoj mreži. Windows 98 SE su također popravili mnoge bugove i softverske probleme koje su imali Windowsi 98 [1].



Slika 7: Windows 98

(Izvor: https://en.wikipedia.org/wiki/Windows_98)

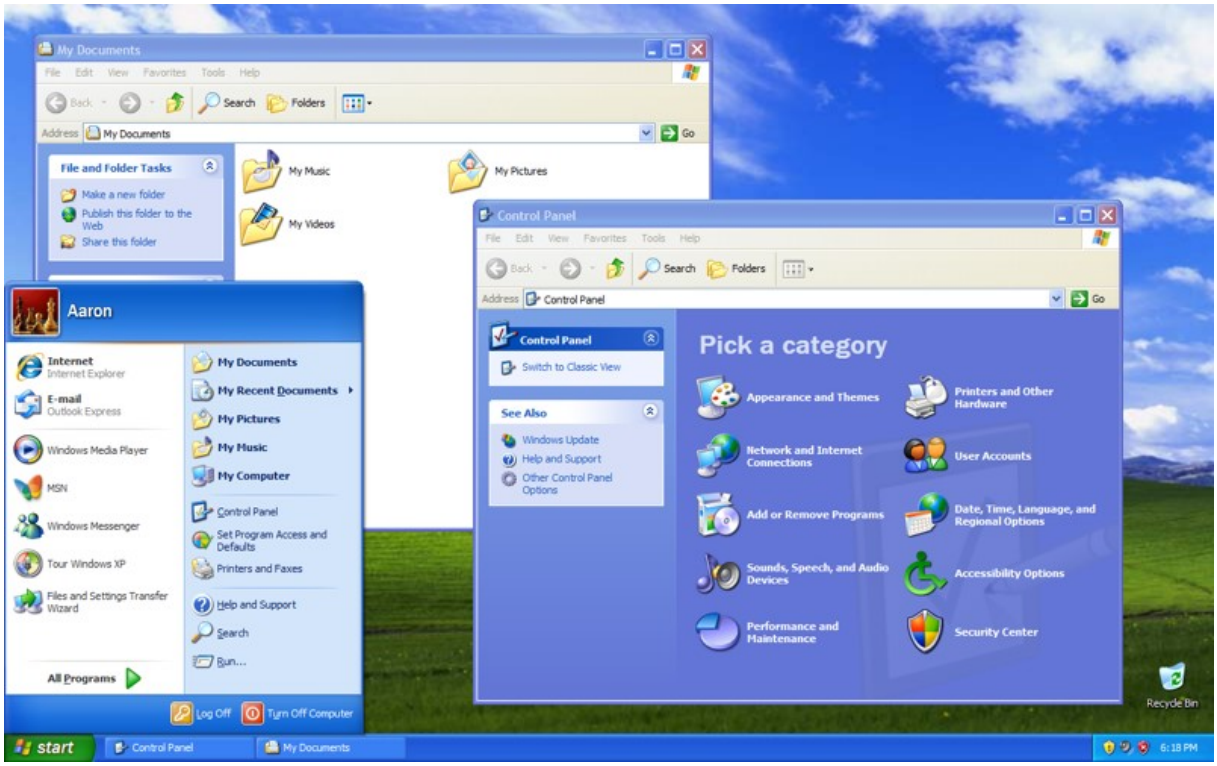
Windows 2000 su nastali 2000. godine iz Windows NT serije. Imali su veću podršku za standardne mrežne tehnologije kao što su DNS, LDAP i Kerberos. Uključivali su i mnoge značajke Windowsa 98 kao što su: Device Manager, Windows Media Player i DirectX koji je i dan danas standard za moderne video igre. Windows 2000 su zadnja verzija Windows NT-a koja nema Product Activation (tj. sustav zaštite od ilegalnog kopiranja operacijskog sustava). Windowsi ME (Millennium Edition) su se temeljili na Windowsima 98, a uključivali su poboljšanu podršku za multimediju i Internet. Također su imali i Windows Movie Maker i System Restore. Kao što je već rečeno ovo je zadnja verzija Windowsa iz svoje serije koja nema zaštitu od ilegalnog kopiranja operacijskog sustava. Ispod na slici je prikaz Windows 2000 [1].



Slika 8: Windows 2000

(Izvor: https://en.wikipedia.org/wiki/Windows_2000)

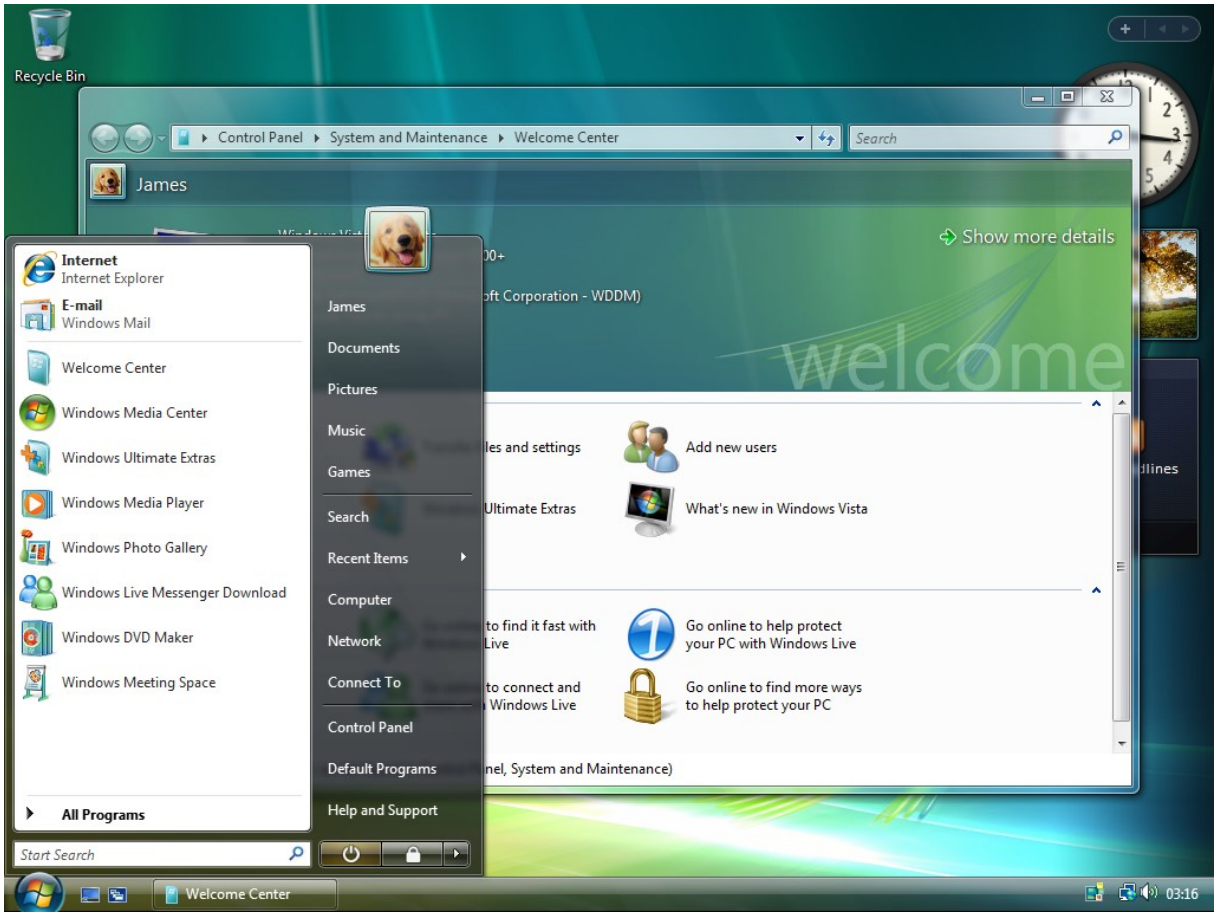
2001. godine nastao je operacijski sustav Windows XP koji su spojili NT/2000 seriju s 95/98/ME serijom. Windows XP koristi Windows NT 5.1 kernel. Poznati su po mnogo sigurnosnih propusta što je rezultiralo sa čak tri veće "Service Pack" nadogradnje koje su uklanjaje te sigurnosne propuste. Service Pack 1 je izašao već 2002. godine, Service Pack 2 je izašao 2004. godine, a Service Pack 3 2008. godine. Nakon Service Pack 2 postali su upotrebljivi širom svijeta za upotrebu. Windows Server 2003 bili su bitna nadogradnja na Windows 2000 Server. Po tvorničkim postavkama imaju isključen zvuk, vizualne teme i grafičko ubrzanje, kako bi radili brže i stabilnije. Ispod je prikaz Windows XP-a [1].



Slika 9: Windows XP

(Izvor: https://en.wikipedia.org/wiki/Windows_XP)

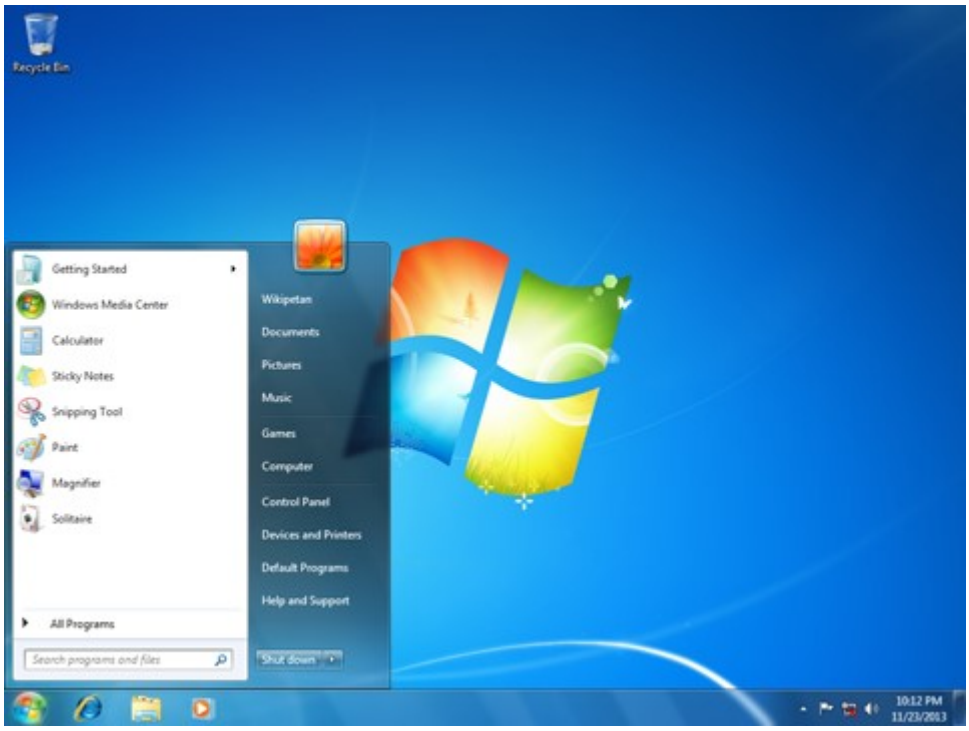
Windows Vista su nastali 2006. godine isključivo za poslovne korisnike te 2007. godine i za širu javnost. Od Windows Viste pa nadalje, novi korisnici više ne postaju automatski administratori. Napravljeni su s namjerom da bi bili sigurniji za uporabu od prethodnih verzija. Imaju novo grafičko sučelje. Sve podverzije osim Starter podverzije imaju 32-bitnu i 64-bitnu varijantu, pri čemu je glavna prednost 64 bitne verzije mogućnost podrške za više od 4GB radne memorije. Windows Server 2008 uključuju sigurnosne pogodnosti iz Viste te su znatno modularniji od svog prethodnika Windows 2003 [5].



Slika 10: Windows Vista 2006

(Izvor: https://en.wikipedia.org/wiki/Windows_2006)

Windows 7 je verzija operacijskog sustava koja je nastala 2009. godine. Karakteristike Windowsa 7 su bili: brže podizanje sustava, bolje upravljanje energijom za prijenosna računala, nadograđeno korisničko sučelje i razna druga poboljšanja koja pojednostavljuju rad s računalom. Na slici ispod je prikaz desktopa i star menu-a Windows 7.



Slika 11: Windows 7

(Izvor: https://sh.wikipedia.org/wiki/Windows_7)

Windows 8 prvi put je javno objavljen 1. kolovoza 2012. godine. Dizajn Windows 8 se uvelike razlikovao od prethodnih verzija Windowsa. Početni zaslon je prvo što će korisnik vidjeti kad se prijavi u sustav. Nije imao gumb start, nije bilo klasične radne površine nego tek redovi šarenih pločica koji se stalno mijenjaju. Na slici ispod je prikaz Windows 8 [3].



Slika 12: Windows 8

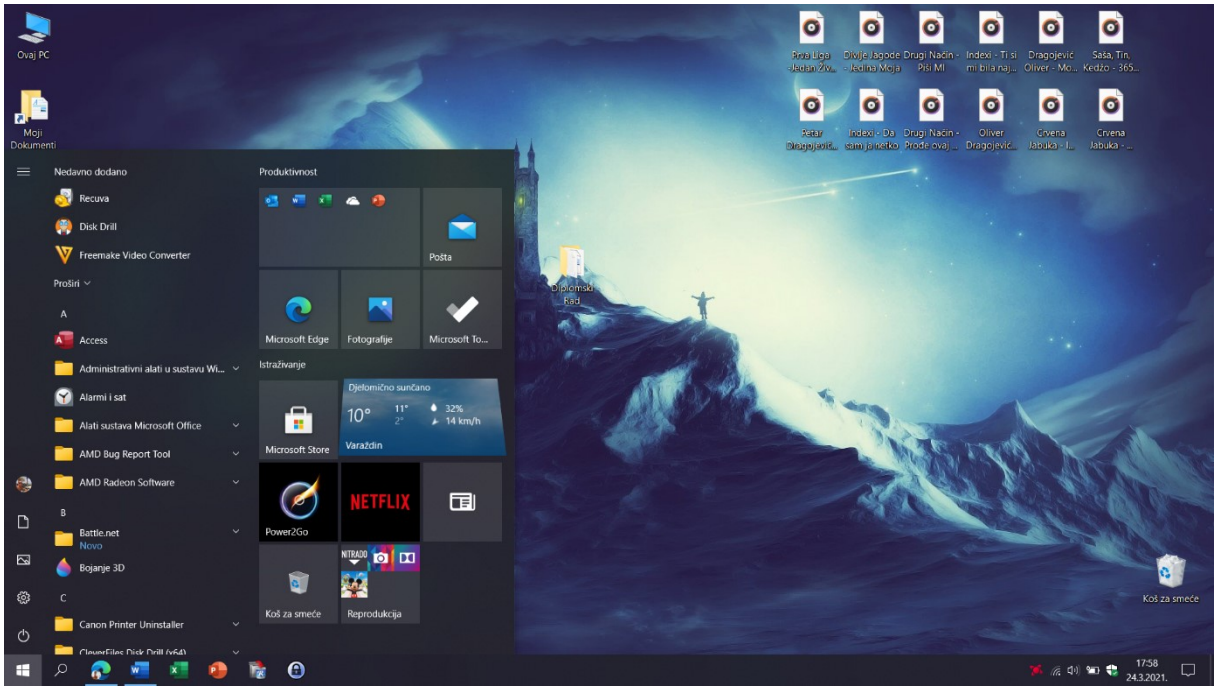
(Izvor: https://www.maketecheasier.com/can-should-i-still-use-windows-81/)

Windows 10 prvi put je javno izdan 29. srpnja 2015. godine. Windows 10 je najnovija inačica Windowsa koja se u najvećoj mjeri koristi. Windows 10 je ujedno i najzahtjevniji operacijski sustav dosad. Sustav se povremeno ažurira na više verzije što omogućuje bolji rad i zaštitu. Na slici ispod je prikaz specifikacija za Windows 10.

Komponenta	Minimalno	Preporučeno
Procesor	1 Ghz IA-32 (32-bitni) ili x64 (64-bitni) Podrška za SSE2, PAE i NX	x64 SLAT
RAM	**32-bitni**: 1 GB **64-bitni**: 2 GB	4 GB
Tvrdi disk	**32-bitni**: 16 GB **64-bitni**: 20 GB	nema preporuke
Grafička kartica	DirectX 9 WDDM 1.0	DirectX 10
Zaslon	1024x600 piksela	1024×768 piksela
Ulazni uređaji	tipkovnica i miš	Multitouch zaslon

Slika 13: Specifikacije potrebne za rad Windows-a 10 na računalu

(Izvor: https://hr.wikipedia.org/wiki/Windows_10)



Slika 14: Windows 10

4.4. Kratka povijest Linuxa

Linux operacijski sustav prvi put je javno izdan u rujnu 1991. godine. Finski znanstvenik Linus Benedict Torvalds tada 21-godišnji student računarstva na Sveučilištu u Helsinkiju je započeo raditi na tom operacijskom sustavu.
Linus Benedick Torvalds 25. travnja 1991. izjavio sljedeće:

„I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since April, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash (1.08) and gcc (1.40), and things seem to work. This implies that I'll get something practical within a few months [...] Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable [sic] (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(. [6]*“*

U samim počecima operacijski sustav Linux je služio kao eksperimentalni sustav koji su tada koristili studenti, programeri, razni hakeri i ljudi koji su bili orijentirani na rad s računalima, nije bilo široke komercijalne upotrebe. Nastankom Apache Web Servera sve se promijenilo, zajedno s Linuxom pružio je pouzdano i besplatno rješenje za postavljanje velikog broja web stranica. Na taj je način Linux u samo nekoliko godina postao velika konkurancija mnogim drugim sustavima koji su bili slični Unixu te u velikoj mjeri i Windows NT-u na serverskom tržištu [4].
Poznatije verzije Linuxa:

- 17. rujna 1991. godine - Linux 0.01 (Verziju 0.01 još nije bilo moguće koristiti jer je njen kod zahtijevao Minix za kompajliranje i pokretanje).
- 5. listopada 1991. godine – Linux 0.02
- Ožujak 1994. godine - Linux 1.0.0. (te je to ujedno bila i prva verzija spremna za korištenje).
- 1996. godine - Linux 2.0.0.
- 2011. godine – Linux 3.0.0.
- Travanj 2015. godine - Linux 4.0.
- Lipanj 2015. godine - Linux 4.1. [6]

„Danas je Linux sustav koji podržava Alpha, ARM, Sparc, x86, X86_64, PPC i ostale arhitekture te dominira na tržištu servera, superračunala i embed uređaja [4]*.“* Linux pogoni uređaje veličine ručnog sata i ogromne data centre. Današnji komercijalan interes za Linux je

poprilično jak. Kompanije poput Novella ili pak divovi poput IBM-a pokazuju popriličan interes za Linux te doprinose njegovom razvoju. Pojavljuje se sve više tvrtki koje nude svoja rješenja za mobilne uređaje, potrošačku elektroniku, desktop računala i servere bazirane na Linuxu ili pak samu korisničku podršku za Linux sustave. Linux je Unixoidan sustav, od Unixa posuđuje mnoge ideje i implementira Unix-API, no nije direktan potomak iz njegova izvorna koda kao BSD.

Jedno od najzanimljivijih svojstava Linuxa je što nije komercijalan produkt već je kolaborativan projekt razvijan preko Interneta. Iako Linus ostaje autor Linuxa i održavatelj kernela, razvoj se održava preko manje grupe developera širom svijeta. Svatko može pridonositi razvoju Linuxa jer Linux je slobodan i otvorena kôda. Točnije Linux je objavljen pod GPL V2 licencom.

„Osnova sustava baziranog na Linuxu je kernel, GCC (GNU C Compiler), C library, toolchain I shell. Današnje moderne distribucije nude i X Window System uključujući i punokrvne desktop managere kao KDE ili pak GNOME uz mnoštvo GNU aplikacija te je korisnicima dostupno na tisuće, što slobodnih što komercijalnih, aplikacija za naknadnu instalaciju“ [4].



Slika 15: Linus Benedict Torvalds

(Izvor: https://tadviser.com/index.php/Person:Torvalds_Linus_of_B%C3%A9n%C3%A9dicte)

5. Zadaci operacijskog sustava

Računalni sustav je sastavljen od procesora, radnog spremnika, vanjskih spremnika i različitih ulazno-izlaznih naprava, sve te komponente čine sklopovlje računala. Različiti primjenski programi, s pomoću kojih korisnici računala obavljaju sebi korisne zadatke, transformiraju računalni sustav u odgovarajući virtualni stroj. Svim tim raznovrsnim programima služi skup osnovnih programa koji omogućuju provođenje operacija računala. Taj skup programa naziva se operacijski sustav. Operacijski sustav nadoknađuje sva ograničenja i nedostatke sklopovlja. Korisnik ne mora biti svjestan detalja provođenja neke operacije koju je on zatražio. Operacijski sustav skriva od korisnika njemu nevažne detalje izvođenja neke operacije. Svrha operacijskog sustava je ta da olakša uporabu računala. Drugi važan zadatak operacijskog sustava je taj da organizira djelotvorno iskorištavanje svih dijelova računala. Operacijski sustav se mora pobrinuti da podržava višeprogramski rad. Mora omogućiti svakom pojedinom programu pristup do potrebnih mu datoteka i svih ostalih sredstava. Sve operacije koje operacijski sustav omogućuje se moraju na neki način moći pokrenuti. Neke zahtjeve postavlja čovjek dok neke dolaze neposredno iz programa. Korisnik može zahtijevati da se neki programi pokrenu ili da se datoteka premjesti s diskete na disk ili obrnuto [7].

Operacije operacijskog sustava može pokrenuti:

- Čovjek preko korisničkog sučelja
- Program preko sučelja primjenskog programa

Operacijski sustav preko tih sučelja povratno vraća rezultate zatraženih operacija ili samo informacije o obavljenom zahtjevu. Najčešće operacije koje pokreće čovjek odnose se na datoteke [7].

6. Von Neumannov model računala

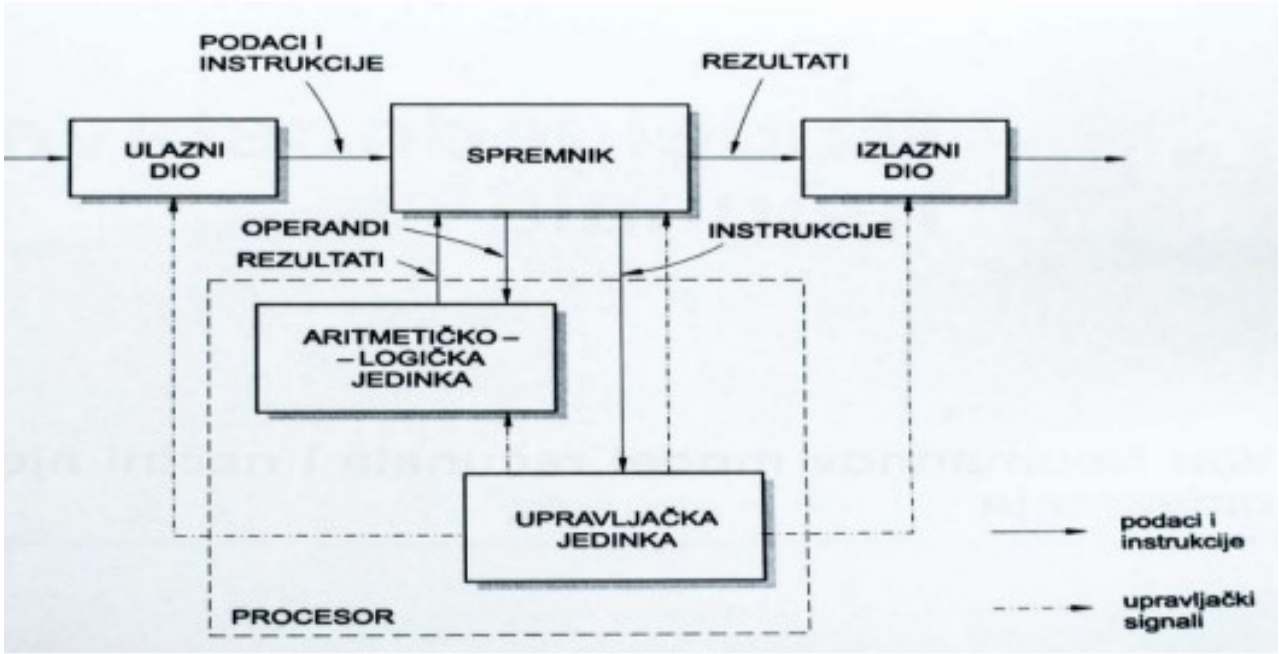
6.1. Funkcijski model računala

Svi današnji računalni sustavi osnovani su još uvijek na koncepcijskom modelu mađarskog matematičara i fizičara Johna von Neumanna iz 1945. godine.

Funkcijski 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 instrukcije programa uneseni izvana, kao i rezultati djelovanja instrukcija.
- Aritmetičko-logička jedinica koja može izvoditi instrukcijama zadane aritmetičke i logičke operacije.
- Upravljačka jedinica koja dohvaća instrukcije iz spremnika, dekodira ih i na temelju toga upravlja aritmetičko-logičkom jedinicom te ulaznim i izlaznim dijelovima [7].

Slika ispod prikazuje međusobnu povezanost svih gore nabrojanih dijelova.



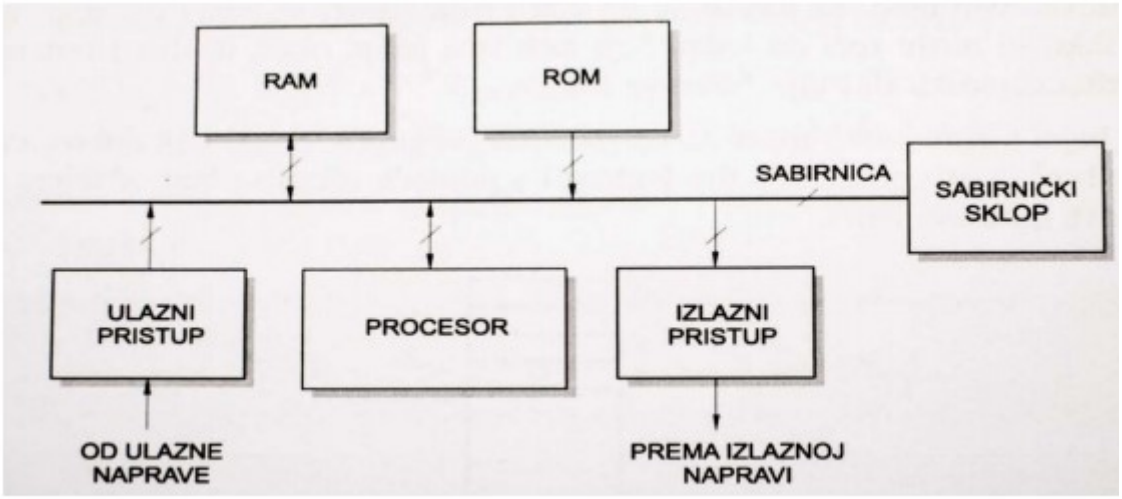
Slika 16: Von Neummanov model računala

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 10))

Nakon što je ukratko opisano kako izgleda funkcijski model računala Johna von Neummana pozornost će se posvetiti ulazno-izlaznim napravama. Objasniti će se spajanje ulazno-izlaznih naprava računala, prijenos znakova računala te prekidni rad procesora.

6.2. Sabirnička građa računala

U funkcijskom prikazu računala vidi se kako su međusobno povezani pojedini dijelovi računala. Svaka crtica koja predstavlja tok podataka, instrukcija ili upravljačkih signala sastoji se od većeg broja vodiča preko kojih se prenose električni signali pomoću kojih se prenose bitovi. Takvo međusobno povezivanje pojedinih dijelova računala nije prikladno 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. Na slici ispod prikazano je računalo sa sabirničkim povezivanjem dijelova [7].



Slika 17: Prikaz sabirničkog povezivanja dijelova računala

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 11))

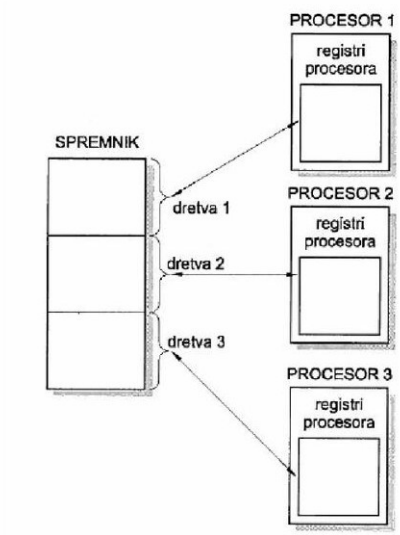
Sve potrebne razmjene podataka, instrukcija i upravljačkih signala obavljaju se preko zajedničkih vodiča. Kosa crtica prikazuje sabirnicu i označava da je riječ o snopu vodiča. Jasno je da se preko zajedničkih vodiča ne može uspostavljati više istovremenih veza već se mogu obavljati samo pojedinačni prijenosi [7].

6.3. Računalni proces

Program koji je zapisan na papiru ili je pohranjen u spremniku računala je niz instrukcija koje procesor mora razumijeti i znati obaviti. Kada se program počinje izvoditi, njemu se pripisuju sljedeća svojstva:

- Trenutak početka izvođenja programa
- Trenutak završetka izvođenja programa
- Trajanje izvođenja programa
- Zaustavljanje izvođenja programa i slično

Na taj način izvođenje programa dobiva obilježje procesa. Prije početka nekog procesa u računalu se stvaraju uvjeti za njegovo provođenje. Zadatak operacijskog sustava je da stvori uvjete za odvijanje programa u kojima se izvođenje može opisati modelom koji ima opis rudimentarnog računala. Stoga u stvarnim računalnim sustavima ponašanje programa možemo promatrati kao proces. Unutar svakog procesa mora postojati barem jedna instrukcijska dretva. Operacijski sustav mora imati sve uvjete za odvijanje procesa, tj. mora osigurati spremnički prostor i procesor u skladu s modelom rudimentarnog računala. Brzina izvođenja dretve je određena brzinom procesora [7].



Slika 18: Prikaz višedretvenog rada na višeprocesorskom računalu

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 31))

U računalima s više procesora, proces se može podijeliti na više dretvi. Spremnik se tada mora podijeliti na način da svaka dretva dobije svoj dio adresnog prostora iz kojeg pripadajući

procesor dohvaća instrukcije i podatke. Računala kod kojih procesor provlači svoje dretve kroz instrukcije i na taj način olakšava zajedničko obavljanje posla nazivamo računalima koja podržavaju višedretveni rad (engl. *multithreading*). Danas svi višenamjenski računalni sustavi podržavaju višedretveni rad i to zbog jedostavnosti izvedbe primjenskih programa i povećanja učinkovitosti korištenja svih dostpunih resursa sustava [7].

7. Ulazno-izlazne naprave

Kako bi se nadopunio Von Neummanov model potrebno je dodati ulazne i izlazne naprave. Ulazno-izlazne naprave omogućavaju komunikaciju čovjeka s računalom.
Najčešće ulazne naprave svakog osobnog računala su:

- Miš
- Tipkovnica

Najčešće izlazne naprave su:

- Monitor
- Pisač

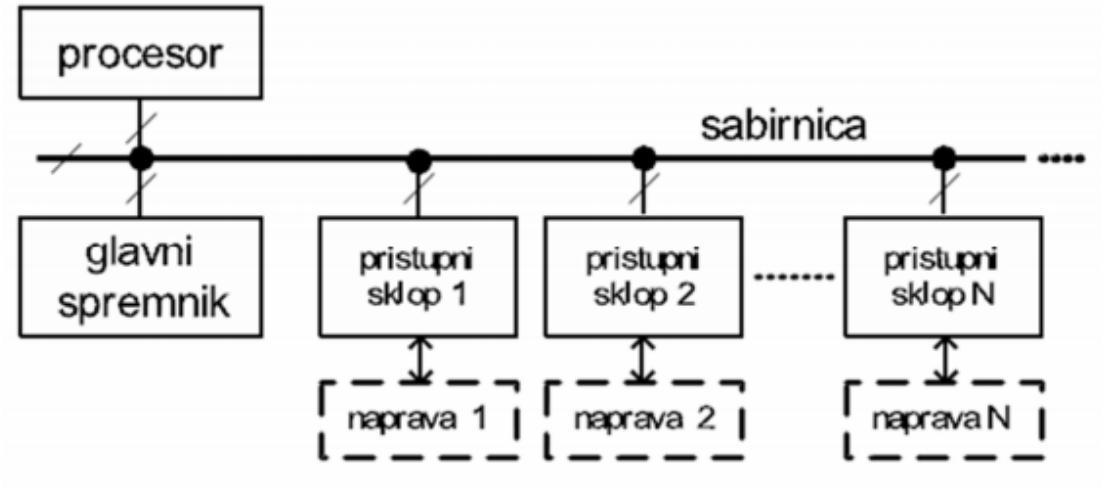
Sve ulazne i izlazne naprave nazivamo zbirnim imenom ulazno-izlazne naprave.
U skupinu tih naprava možemo ubrojiti i vanjske spremnike, koji su i ulazne i izlazne naprave. To su:

- Magnetski diskovi
- Magnetske diskete
- Optički diskovi

Osim ovih osnovnih naprava današnja računala mogu se opremiti raznovrsnim ulazno-izlaznim napravama i uređajima. Tako se na računala mogu priključiti:

- Naprave za čitanje znakova i slika
- Naprave za crtanje i izradu slika
- Videokamera
- Videorekorder
- Kompaktni disk
- Mikrofon
- Zvučnici

Navedene ulazno-izlazne naprave neovisne su o brzini odvijanja programa u procesoru i stoga je potrebno posebnu brigu posvetiti vremenskom usklađivanju rada procesora i ulazno-izlaznih naprava. Svaka ulazno-izlazna naprava ima svoj vlastiti upravljački elektronički sklop koji se brine o svim detaljima. Naprave su spojene na pristupni sklop računala koji je s jedne strane prilagođen toj napravi, a sdruge strane se prilagođava sa protokolima sabirničkog sustava, pristupni sklop mora omogućiti i ostvarenje sinkronizacije. Na slici ispod je ilustracija načina povezivanja neke ulazno-izlazne naprave na sabirnicu računala.



Slika 19: Prikaz spajanja ulazno-izlaznih naprava na sabirnicu računala

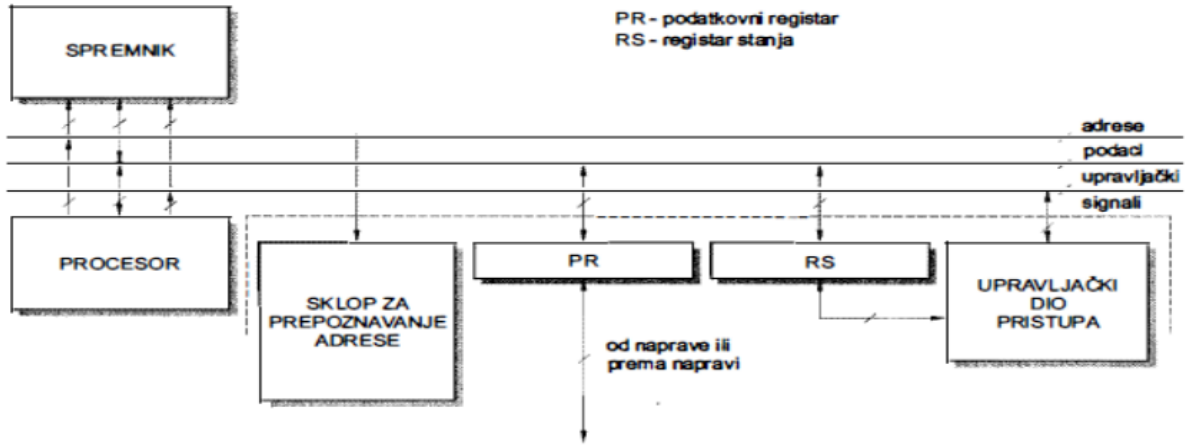
(Izvor: Nino Dasko (2015.) Operacijski sustavi.)

Naprave i njihove pristupne sklopove možemo podijeliti u dvije skupine:

- Naprave kod kojih se prenose blokovi znakova
- Naprave kod kojih se prenose pojedinačni znakovi

7.1. Prijenos pojedinačnih znakova

U nastavku prikazan je shematski detaljni shematski prikaz pristupnog sklopa za prijenos pojedinačnih znakova. Prikazana su dva registra: podatkovni registar (njega možemo promatrati kao jedan bajt spremnika) i registar stanja (u njegove bitove se mogu pohraniti informacije o stanju pristupnog sklopa, kao i upravljački bitovi koji mogu modificirati ponašanje pristupa). Procesor može te registre adresirati jednako kao i što adresira pojedine bajtove spremnika. Sklop za prepoznavanje adrese određuje koji je od registra adresiran, a posebni upravljački signal određuje je li se taj adresni registar piše ili čita.



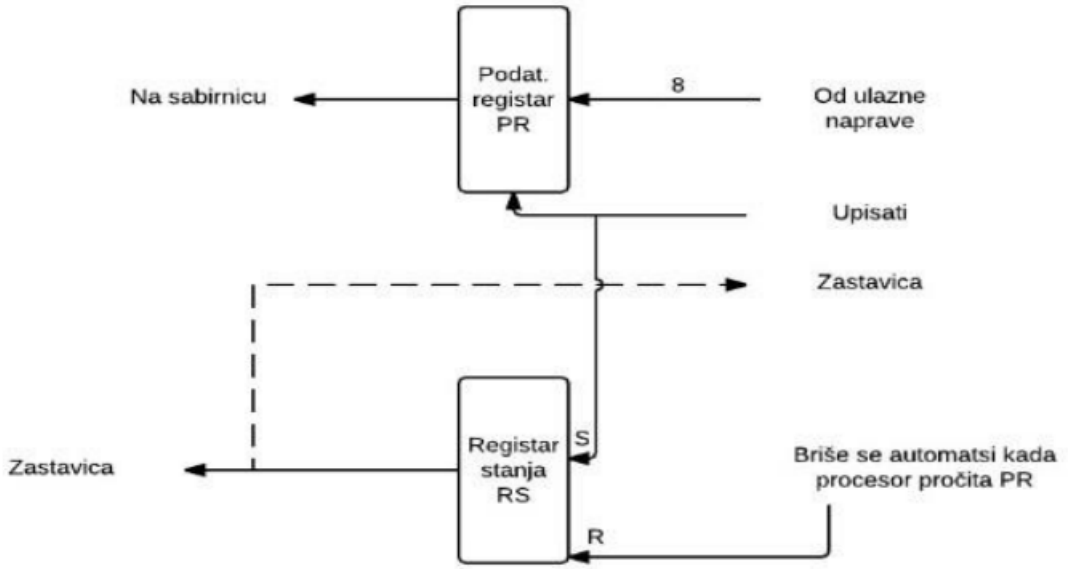
Slika 20: Shematski prikaz pristupnog sklopa za prijenos pojedinačnih znakova

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 35))

Za vrijeme obavljanja izlazne operacije, to jest prilikom prijenosa sadržaja nekog bajta spremnika prema izlanom uređaju procesor mora obaviti iduće operacije:

- Jednom instrukcijom dobaviti taj sadržaj iz spremnika u svoj registar
- Drugom instrukcijom prenijeti taj znak u podatkovni registar pristupnog sklopa

Kada govorimo o pristupnom sklopu i upravljačkom sklopovlju naprave oni su dužni osigurati da se sadržaj prenese dalje prema izlaznoj napravi. Za vrijeme prenošenja znakova, procesor ne smije staviti sljedeći znak u podatkovni registar pristupnog sklopa ako znak nije još potrošen. U registru stanja izlazna zastavica predstavlja posebni bit i ona ostaje postavljena onoliko dugo dok znak ne bude prihvaćen od strane izlazne naprave. Na sličan način se obavlja i ulazna operacija. Na slici ispod prikazan je prijenos znaka u radni spremnik.



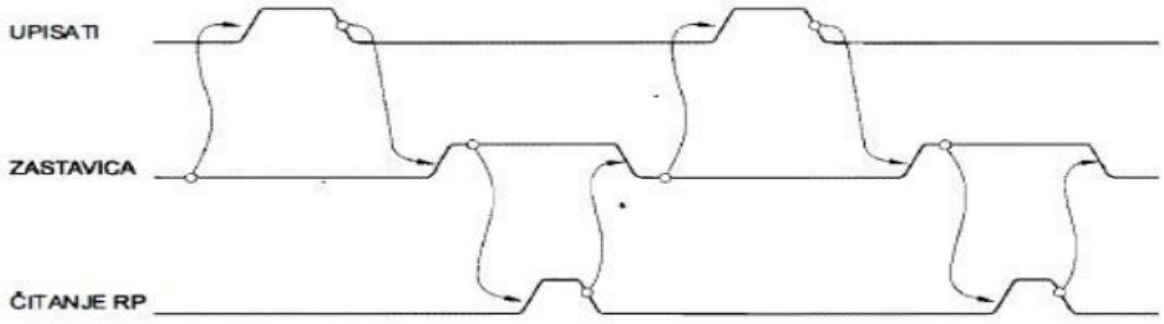
Slika 21: Shematski prikaz ulaza pojedinačnog znaka

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 36))

Što se tiče ulaza pojedinačnog znaka, od ulazne naprave u pravilu dolazi osam vodiča s kojim dolazi osam bitova jednog bajta. Ti bitovi se upisuju u podatkovni registar (PR) onda kada upravljački sklop uređaja posebnim signalom to naredi. S upisom novog sadržaja u podatkovni registar se postavlja bit koji je označen kao zastavica, ta se zastavica pretvara u električmni signal i vraća se nazad prema ulaznom uređaju. Upravljački sklop mora postaviti signal „upisati“ samo u trenutku kad je zastavica spuštena. Tim mehaniznom stvorena je sinkronizacija sa strane ulazne naprave. Na taj način će ulazni uređaj pohraniti novi znak u podatkovni registar samo onda kada je zastavica spuštena i onda će se automatski pri tom upisu podignuti.

Procesor uzastopno čita registar stanja (RS) pristupnog sklopa toliko dugo dok se ispitivanjem bita zastavica ne ustanovi da je on postavljen. Nakon što je to obavljeno treba pročitati podatkovni registar (PR). Prijenosom sadržaja registra (PR) u jedan od registra procesora, pristupni sklop postaje spreman da prihvati novi znak. Upravljački dio pristupnog sklopa pri čitanju registra (PR) automatski spušta bit zastavicu na nulu te je to označeno oznakom R uz bit zastavica.

Vodiče koji su nazvani „upisati i „zastavica“ moguće je vidjeti na izvodima pristupnog sklopa (PR) i na njima se nalaze električni signali. Ako bi smo spojili osciloskop na njih dobili bi promjene napona koje su prikazane na slici ispod.



Slika 22: Prikaz protokola dvožičnog rukovanja

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 37))

Na slici iznad prva dva reda predstavljaju nam promjene napona. Strelice na slici prikazuju uvjetovanost, tj. Što se točno zbiva. Krugovi na niskim razinama signala „zastavica“ označava da ulazna naprava ne smije generirati signal „upisati“. Krug na padajućem dijelu signala „upisati“ uzrokuje podizanje signala „zastavica“, dok podignuta zastavica predstavlja preduvjet za čitanje registra (PR). Čitanje RP nam predočava pojavu čitanja registra podataka te uzrokuje spuštanje signala „zastavica“.

7.2. Prijenos blokova znakova neposrednim pristupom spremniku

Mnoge aktivnosti na računalu povezane su sa prijenosom većeg broja znakova. Na taj način se obavlja:

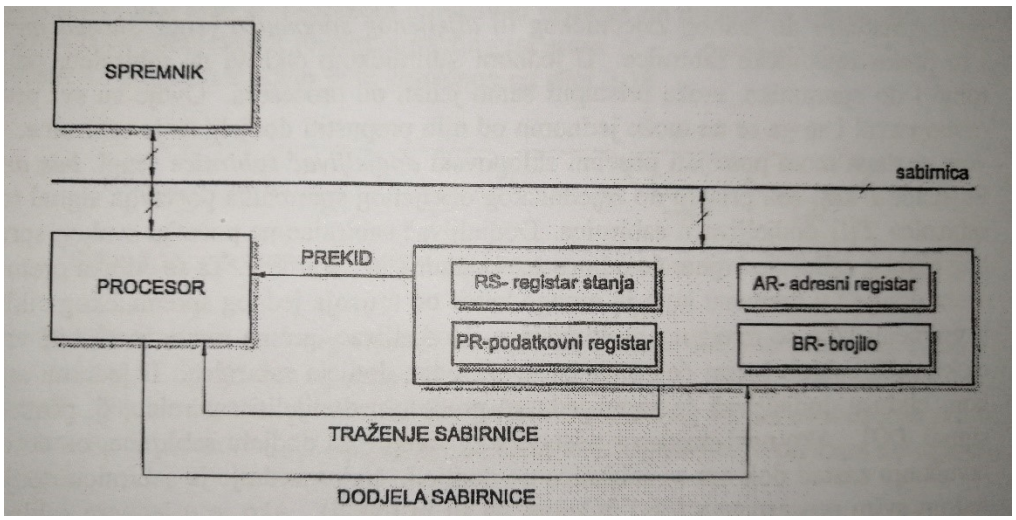
- Prijenos sadržaja iz radnog spremnika na disk i obrnuto
- Prijenos sadržaja iz radnog spremnika u videospremnik pristupnog sklopa monitora
- Prijenos sadržaja prema komunikacijskim pristupnim sklopovima

U prethodnim poglavljima ustanovljeno je da prijenos pojedinačnih znakova bloka u prekidnom načinu rada izaziva mnogo kućanskog posla. Procesor je bivao prekinut pri prijenosu svakoga znaka, a nakon svakog prekida prenosio bi se samo jedan znak. Pritom procesor obavlja vrlo jednostavan posao koji se sastoji od nekoliko instrukcija.

Cijeli taj posao koji radi procesor bi se mogao prepustiti pristupnom sklopu. Da bi to bilo moguće model pristupnog sklopa bi se morao nadopuniti sa još dva registra:

- Adresnim registrom AR u kojem će se nalaziti adresa u spremniku na koju će se obaviti prijenos pojedinog znaka
- Brojilom BR u kojem se nalazi broj znakova koje još treba prenijeti

Dodatna dva registra koja su spomenuta trebaju imati svoje adrese tako da bi ih se moglo programski dohvatiti. Ispod je shema takvog pristupnog sklopa (PR).



Slika 23: Shema jednostavnog modela za neposredni pristup spremniku

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 54))

Prijenos bloka znakova programski se započinje tako da procesor zapiše početnu adresu u adresni registar (AR) u spremnik i u registar brojilo (BR) broj znakova koji se želi prenijeti. Nakon toga se upisom odgovarajućeg sadražaja u registara stanja (RS) pokreće prijenos bloka.

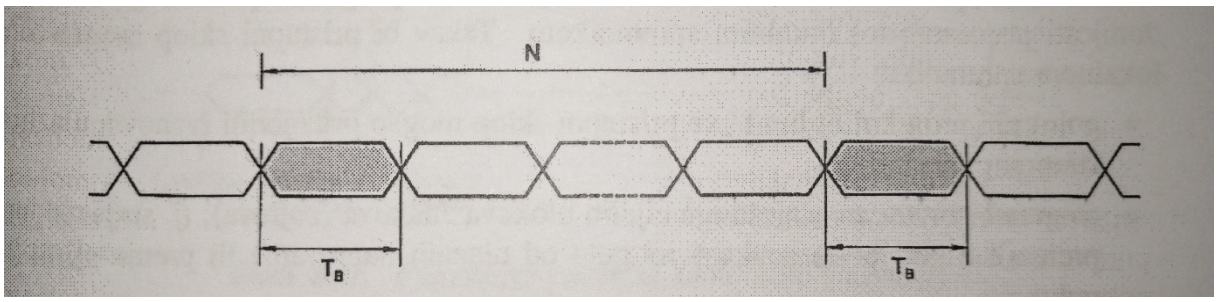
Unutar pristupnog sklopa (PR) obavlja se sljedeći programski kod:

```
dok je (BR > 0){
     zatražiti sabirnicu;
     čekati na dodjelu sabirnice;
     postaviti na adresni dio sabirnice sadržaj registra AR;
     prenijeti na tu adresu sadržaj podatkovnog PR (ili obrnuto);
     AR = AR + 1;
     BR = BR - 1;
}
postaviti signal PREKID;
```

Slika 24: Programski kod pristupnog sklopa

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 55))

Prekid procesora dogodit će se nakon što se prenese cijeli blok. Obrada prekida sastoji se od ponovne inicijalizacije registara adresnog bloka (AR) i brojila (BR) ako se želi nastaviti s prijenosom sljedećeg bloka. Odvijanje prijenosa cijeloga bloka između dvaju prekida obavlja se bez sudjelovanja procesora. Na slici ispod prikazana je shema prepuštanja spremničkog ciklusa pristupu.

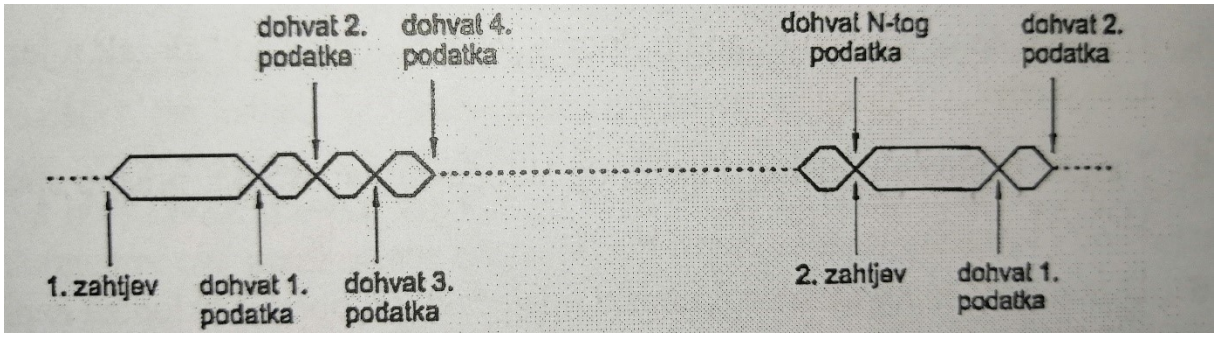


Slika 25: Shema prepuštanja spremničkog ciklusa pristupu

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 55))

7.3. Sabirnički sustav stvarnih računala

Stvarna računala podržavaju prethodno prikazane načine rada sabirnice, ali većinu vremena to rade na drugačiji način. Frekvencija rada procesora je najbrža komponenta računalnog sustava. Komponente kao što su spremnik i grafički podsustav rade usporedivim frekvencijama, tj. nekoliko puta nižim frekvencijama. Zbog toga se osim jedne brze sabirnice u sustav uvodi jedna sporija sabirnica koja je s brzom sabirnicom povezana preko prikladnih premosnika (engl. bridge). Na sporiju sabirnicu spojene su sporije komponente računalnog sustava kao ulazno izlazni podsustavi. Isto tako ni jedna takva podjela nije dovoljno učinkovita za iskorištavanje mogućnosti procesora koji je i dalje puno brži od ostalih brzih komponenti sustava. Zato se intezivno koriste priručni spremnici procesora.



Slika 26: Prikaz rada priručnih procesora

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 59))

8. Programi, procesi i dretve

8.1. Općenito o procesima

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

Kada se program preveden u strojni oblik pokrene, on dobiva neke vremenske atribute, kao što su:

- *Trenutak početka*
- *Trajanje izvođenja*
- *Trenutak završetka*

Tijekom svog izvođenja programa može zastati i biti ponovno pokrenut. Prema tome, program koji se izvodi djeluje tako da na uredni način utječe na promjene stanja računalnog sustava“

Više procesa može istodobno napredovati ako se odvijaju u različitim dijelovima računalnog sustava. Primjerice ako jedan proces čeka završetak ulazne operacije kako bi mogao nastaviti neko izračunavanje za to vrijeme drugi proces može izvoditi svoje instrukcije [7].“

8.2. Model višedretvenosti

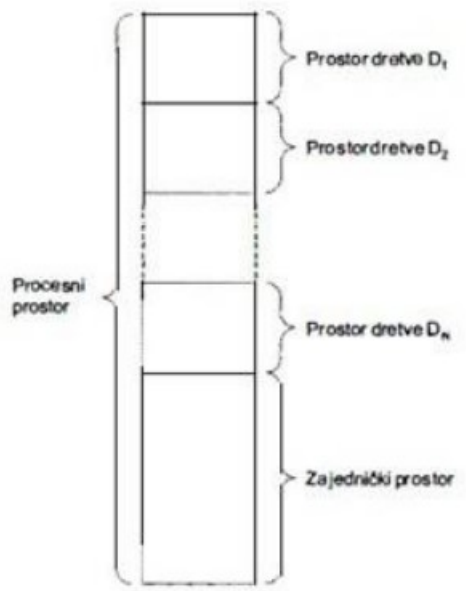
Pretpostavimo da se sve dretve nalaze unutar istog procesa. Nakon toga se sve dretve dijele na dretvin spremnički prostor i svaka dretva ima:

- skup instrukcija
- skup podataka
- vlastiti stog

Nadalje svim dretvama procesa na raspolaganju stoji adresni prostor procesa. Svaka individualna dretva zauzima svoj dio spremničkog prostora. Taj prostor podijeljen je na:

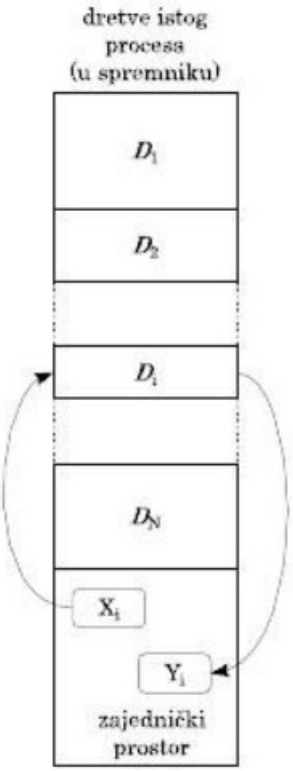
- dio u koji su smještene instrukcije dretve,
- dio u kojem je smješten stog dretve,
- dio u kojem su smješteni lokalni podaci dretve.

Procesni adresni prostor kojeg mogu koristiti sve dretve, sastoji se od zajedničkog prostora u koji se mogu smjestiti zajedničke varijable svih dretvi. Takav adresni prostor, prikazan je na slici 25. Razmatranjem višedretvenosti može se pretpostaviti da je u sustavu osigurana pravilna promjena konteksta dretvi i da dretve mogu nesmetano izvoditi, prekidati i nastavljati. Shvaćanje ponašanja dretvi može se jednostavnije shvatiti kroz jednostavan primjer. „Pretpostavka je da dretva D1 kreće s čitanjem podataka iz podskupa koji se nalazi u zajedničkom dijelu spremnika, odnosno iz svoje domene, koju možemo nazvati Xi“. Na kraju čitanja podataka, dretva će pisati svoje rezultate u jedan podskup koji se nalazi u zajedničkom dijelu spremnika (u svoju kodomenu nazvanu Yi). Iz navedenog može se zaključiti da dretva nakon obavljene obrade podataka, koristeći samo svoj dio adresnog prostora, završava i zapisuje rezultate u svoju kodomenu [7].



Slika 27: Prikaz modela višedretvenosti

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 64))



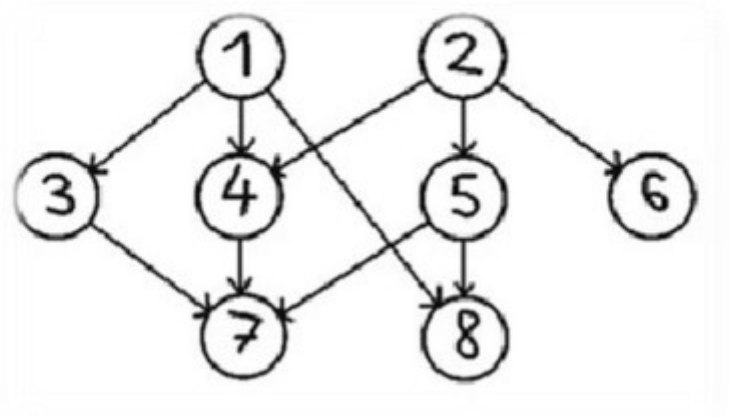
Slika 28: Dretva, domena i kodomena

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 65))

„Dva podzadatka jednog zadatka ne moraju biti međusobno povezana. Dva su pozadatka međusobno nezavisna ako nemaju zajedničkih lokacija u svojim domenama i kodomenama, te se ona mogu izvoditi bilo kakvim redoslijedom. U drugom slučaju, podzadaci mogu ovisiti jedan o drugome. Redoslijed izvođenja postaje bitan ukoliko jedan podzadatak čita svoje ulazne podatke iz lokacija u koji neki drugi podzadatak piše svoje rezultate [11].“

8.3. Sustav Dretvi

Programski zadaci se u pravilu sastoje od više podzadataka te se u takvom sustavu mora za svaki par zadataka utvrditi zavisnost ili nezavisnost. Na taj način se utvrđuje redoslijed izvođenja. Sustav podzadataka pretvorit će se u sustav dretvi te će zajedničkim djelovanjem obaviti zadani zadatak, ako se zadatak može rastaviti na više podzadataka tada se ramatranjem njihovih domena i kodomena mogu utvrditi međusobno zavisni i nezavisni zadaci te se sustav može prikazati usmjerenim grafom. Na slici 29. prikazan je primjer sustava podzadataka [8].



Slika 29: Grafički prikaz podzadatka

(Izvor: Leonardo Jelenković: Materijal za predavanja: Operacijski sustavi, 2014.)

Na prikazanom primjeru iznad može se vidjeti kako se dretva (1) mora obaviti prije dretve (3), dretve (4), dretve (8) te dretve (7). Dretve koje se nalaze na istom putu međusobno su zavisne i moraju se izvoditi točno određenim redoslijedom. Dok one dretve koje se nalaze na različitim putevima one su nezavisne i mogu se izvoditi paralelno. Nezavisne dretve mogu biti izvođene istodobno ako na raspolaganju imaju više fizičkih procesora ili prividno istodobno ako se izvode na istom procesoru. *„Kako bi bilo moguće izvesti navedeni posao, odnosno kako bi se omogućilo pokretanje, zaustavljanje i komunikacija među dretvama, potrebno je imati mehanizme sinkronizacije. Također, međusobno zavisne dretve moraju imati nekoliko načina razmjene podataka. Jedan od najjednostavnijih načina jest taj da jedna dretva neposredno piše u domenu druge dretve“ [11].*

8.4. Međusobno isključivanje dretvi

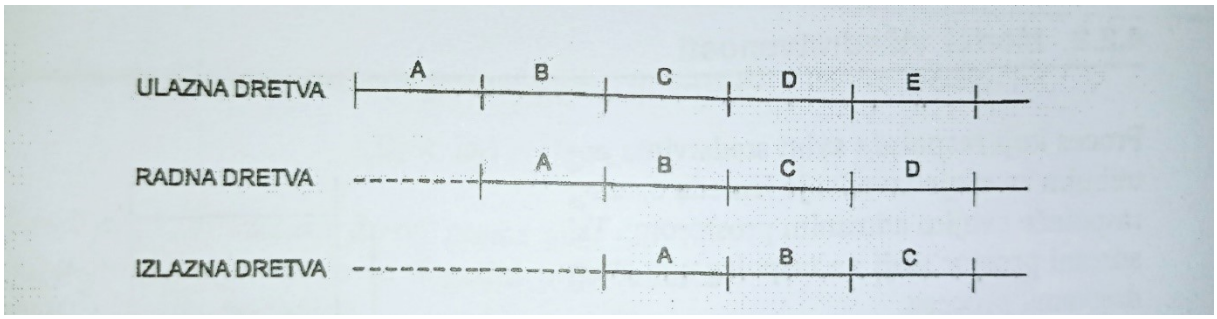
Za dretve je važno da imaju metodu pomoću koje se mogu isključiti. Mehanizam isključivanja dretvi osigurava da se sredstva računalnog sustava koriste pojedinačno. Kritičnim odsječcima nazivaju se dijelovi dretvi koji koriste neko zajedničko sredstvo. Nezavisne dretve smiju prolaziti kroz svoj odsječak samo pojedinačno, kada se dretve izvode u svom ne kritičkom odsječku tada se mogu izvoditi proizvoljnom brzinom i redoslijedom. Isto tako kada dretva obavi svoj kritički odsječak, ona može programski omogućiti prekid i time prelazi u nekritički odsječak.

Međutim problemi se javljaju u višeprocesorskom sustavu jer zabrana prekidanja jednom procesu ne rješava problem. Promatranjem ostvarenja međusobnog isključivanja pretpostavit ćemo da se dretve u svom izvođenju ponavljaju, da se izvode ciklički što će biti objašnjeno detaljnije u idućem poglavlju.

8.5. Cikličke Dretve

U računalnim sustavima mnogo operacija se iznova ponavlja. U najvećoj mjeri to se odnosi na funkcije koje obavlja operacijski sustav. „Cjevovodno ponašanje računalnog sustava na slici 30 ispod moglo bi se postići tako da se:

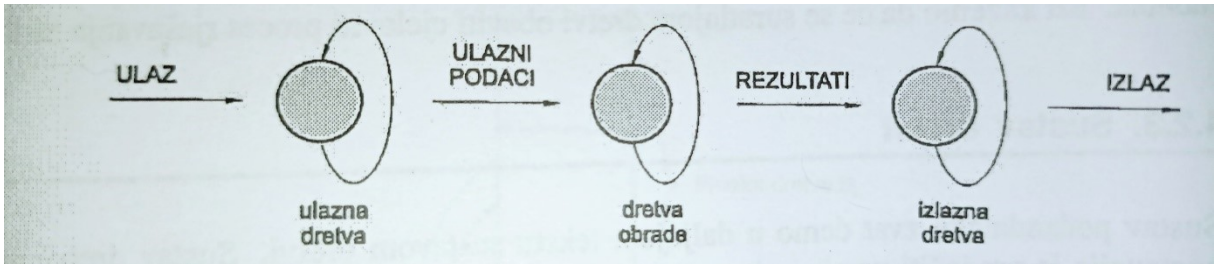
- Ulazna dretva uzastopce ponavlja prihvaćajući u svakom prolazu kroz petlju ponavljanja novu skupinu podataka i prenoseći ih dretvi obrade
- Radna dretva prihvaća podatke koje joj daje ulazna petlja, obrađuje ih te predaje rezultate izlaznoj dretvi i vraća se na početak, gdje čeka nove podatke
- Izlazna dretva prihvaća rezultate radne dretve, prosljeđuje ih u izlaznu napravu i vraća se na svoj početak, gdje čeka nove rezultate“



Slika 30: Cjevovodno ponašanje trodretvenog zadatka

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 63))

„Sve se te dretve trajno ponavljaju i možemo ih nazvati i cikličnim dretvama. U slikovnom prikazu to ćemo ponavljanje istaknuti granom koja uz čvor čini petlju, kao što je to prikazano slikom ispod.“



Slika 31: Ciklične dretve u ostvarenju cjevovodne obrade

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str.69))

9. Jezgra operacijskog sustava

Jezgra operacijskog sustava je centralni dio svakog modernog operacijskog sustava i ona mora postojati. Za jezgru se može reći kako ona objedinjuje i povezuje sve mehanizme koji su navedeni u prethodnim poglavljima. Glavna zadaća jezgre jest da upravlja sklopovljem na najnižoj mogućoj razini te da upravlja različitim procesima i programima. Jezgra operacijskog sustava upravlja i računalnim resursima te predstavlja sloj između fizičkog računalnog sklopovlja i korisničkih programa. Takav sloj nazivamo i jezgrenom okolinom, a razlog postojanja takvog sloja je u dizajnu velike većine modernih operacijskih sustava.

Jezgra operacijskog sustava izvršava se u nadglednom načinu rada što znači da ima ovlasti nad svim sklopovljem računala. Jezgra operacijskog sustava može mijenjati sadržaje registra centralnog procesora te može upravljati prekidima i pristupati ovlaštenim adresnim prostorima. Jezgra operacijskog sustava mora omogućiti izvršavanje više programa, odnosno, procesa pri čemu je također potrebna sinkronizacija procesa, komunikacija među procesima te dijeljenje memorije među procesima. Jezgra operacijskog sustava mora se brinuti da svakom programu bude dodijeljen vlastiti adresni prostor te da se može izvršavati konkurentno s ostalim programima.

Kada govorimo o dretvama, one dretve koje žele komunicirati s drugim dretvama pozivaju određene jezgrine funkcije koje će za njih obaviti zadani posao. Isto je i s izlaznim ili ulaznim operacijama, sve se obavlja preko jezgre operacijskoga sustava. Kada dretve obave svoj posao, jezgra će zaustaviti njeno izvođenje.

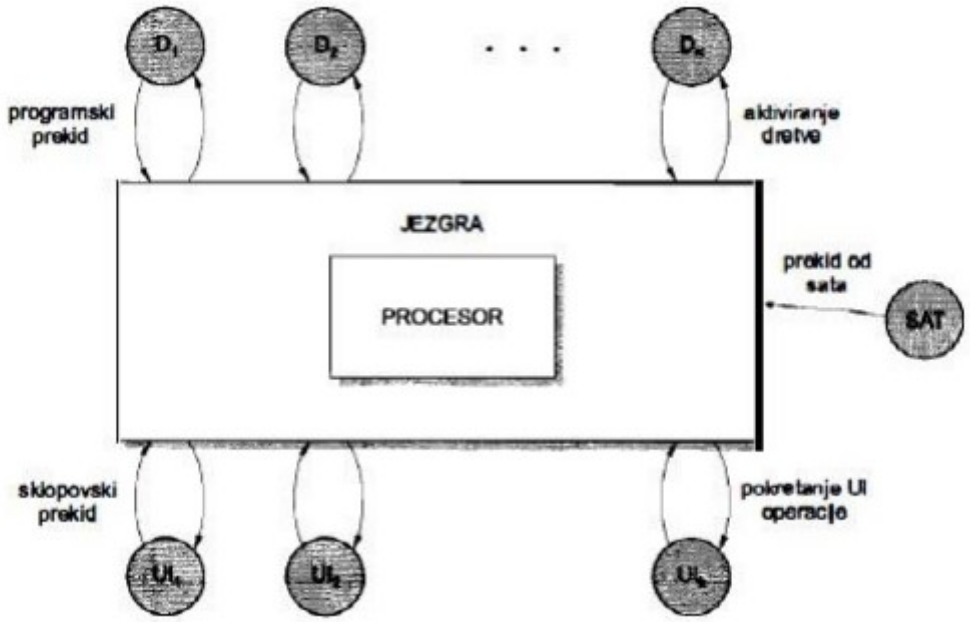
9.1. Jednostavni model jezgre

„Jednostavni model jezgre zasnovan je na idućim pretpostavkama:

- U adresnom prostoru procesa smješteni su svi dretveni prostori
- Čitav adresni prostor procesa dohvatljiv je svim dretvama
- Izvođenje dretvi obavlja se u jednoprocesorskom sustavu
- Dretve odgovarajućim programskim prekidom mogu pozivati jezgrine funkcije
- U sustavu postoje ulazno-izlazne naprave koje se ovdje uzimaju u razmatranje kako bi se jezgrom obuhvatili i obradili sklopovski prekidi„

U sustavu postoje tri vrste prekida i oni su prikazani na slici ispod, a to su:

- Sklopovski prekidi
- Periodni sklopovski prekidi od sata
- Programski prekidi koje aktiviraju dretve



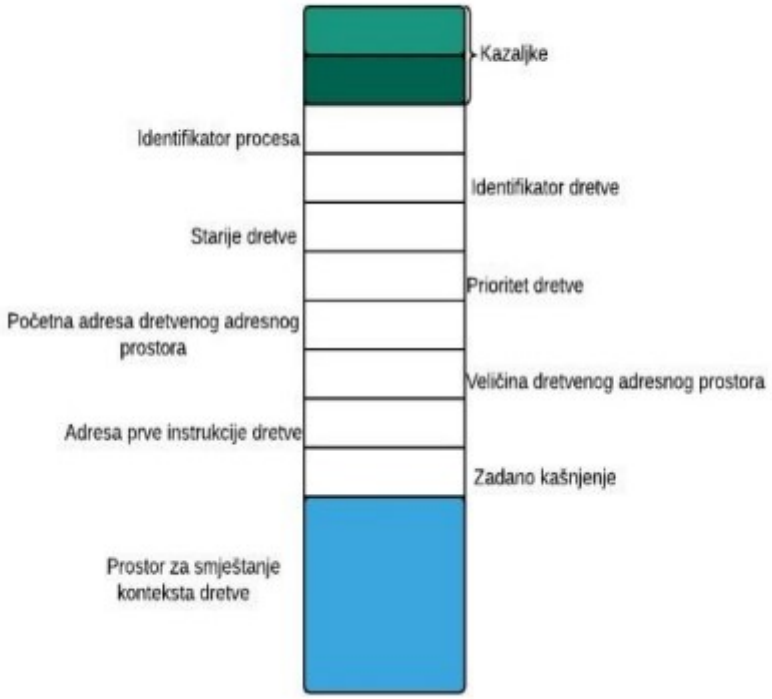
Slika 32: Slikovni prikaz jednostavnog modela jezgre

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 91))

Svi navedeni prekidi služe za pozivanje funkcija jezgre. Kao što je već rečeno prekidom procesor ulazi u sustavski način rada s onemogućenim daljnim prekidanjem. U sustavskom adresnom prostoru nalaze se sve potrebne strukture podataka i programski odsječci koji će ostvariti jezgrine funkcije. Unutar jezgre se donose odluke za daljnje odvijanje dretvi, te odluke se donose u skladu s pravilima o dretvama. Jezgra operacijskog sustava je prikazana na slici 32. Dretve prikazane na slici D1, D2,…Dn su u krugovima. Strelice koje idu od dretve prema jezgri prikazuju pozive jezgrinih funkcija koje su ostvarene programskim prekidima. Strelice koje idu od jezgre prema dretvi govore nam to da dretve mogu biti pokrenute samo djelovanjem jezgre. Ulazno-izlazne naprave prikazane su također u krugovima u kojima su označene sa UI1, UI2,…UIk. Svaka ulazno-izlazna operacija se pokreće iz jezgre operacijskog sustava. Naprava se javlja prekidom kada pokrenuta operacija bude završena. Prekid je prikazan strelicom od naprave prema jezgri, dok je pokretanje UI operacije prikazano obrnuto. Na slici je još prikazan i sat koji prekidom također poziva jezgrinu funkciju. Kod sata postoji samo strelica prema jezgri, zato jer on radi trajno i automatski se pokreće prilikom pokretanja sustava.

9.1.1. Struktura podataka jednostavnog modela jezgre

U ovom poglavlju reći će se više o strukturi podataka jednostavnog modela jezgre i opisati će se koja je uloga pojedinih komponenti te strukture. U jezgru se moraju pohraniti svi važni podaci svake dretve, najbolje bi bilo kada bi se ti podaci smjestili u jedan zapis koji se može nazvati opisnik ili deskriptor dretve.



Slika 33: Sadržaj opisnika dretve

(Izvor: Budin L., Golub M., Jakobović D., Jelenković L. : Operacijski sustavi, 1. Izdanje, Zagreb 2010., (str. 92))

„Na slici 33 prikazan je sadržaj opisnika dretve. Uz mjesta za kazaljke (koja su u opisniku simbolizirana izlaznom strelicom), u opisniku su predviđena mjesta za smještanje parametara:

- *Identifikator procesa označava kojem procesu pripada dretva (prirodni broj).*
- *Identifikator dretve omogućuje međusobno razlikovanje dretvi (prirodni broj).*
- *Stanje dretve je parametar koji označava u kojem se stanju dretva trenutačno nalazi.*
- *Prioritet dretve je parametar koji određuje prednost dretvi pri dodjeljivanju procesora.*
- *Početna adresa dretvenog adresnog prostora i veličina prostora opisuju smještanje dretvenog adresnog prostora unutar procesnog adresnog prostora.*
- *Adresa prve instrukcije sadrži adresu na kojoj je smještena adresa prve instrukcijske dretve.*

- *Zadano kašnjenje je parametar koji će odrediti odgađanje izvošenja dretve za zadani broj perioda otkucaja sata.*
- *Prostor za smještanje konteksta poslužiti će nam za smještanje konteksta dretve onda kada bude dretva prekinuta u svom izvođenju. „*

„*U strukturi podataka jezgre postoje ili se mogu kreirati zaglavlja listi u koje će se svrstavati opisnici dretvi. Pojedine jezgrine funkcije premještat će, prema potrebi, opisnike dretvi iz jedne liste u drugu. Smještanje opisnika dretve u pojedinu od lista odredit će njezino trenutačno stanje [7].“*

9.2 Jezgrine Funkcije

U ovom poglavlju detaljnije će se opisati jezgrine funkcije.

9.2.1. Funkcije za ulaz i izlaz iz jezgre

Ulaz u jezgru događa se pod utjecajem sklopovskog ili programskog prekida. *„U jednoprocesorskom sustavu time je osigurano međusobno isključeno obavljanje jezgrinih procedura“* [7]. Pojavom programskog ili sklopovoskog prekida događa se sljedeće:

- Onemogućuje se prekidanje
- Programsko brojilo smješta se na sustavni stog
- Prekidni podsustav premješta kontekst iz registara procesora na sustavski stog
- Prekidni podustav poziva odgovarajuću jezgrinu funkciju

Funkcija jezgre koja je prekidom pozvana treba na svome početku prvo premjestiti kontekst sa sustavskog sloga u opisnik dretve koja je u trenutku prekida bila aktivna. Opisnik dretve se nalazi u redu Aktivna_D i taj se opis opisuje sljedećeom instrukcijom: pohraniti kontekst u opisnik Aktivna_D [7].

U praktičkim izvedbama pohranjivanje konteksta se smanjuje tako da se pri obradi prekida najprije kazaljka sustavskog stoga preusmjeri na Aktivnu_D i time izbjegne premještanje konteksta. Na samom početku rečeno je kako se izlaz iz jezgre svodi na aktiviranje jedne od dretvi. To će biti dretva koja se nalazi na prvom mjestu u radu Pripravne_D. Aktiviranje se može opisati sljedećim nizom instrukcija:

premjestiti prvi opisnik iz reda Pripravne_D u red Aktivna_D;
obnoviti kontekst iz opisnika Aktivna_D;
omogućiti prekidanje;
vratiti se iz prekidnog načina;

Tablica 1: Prikaz instrukcije za pohranjivanje konteksta

Instrukcija *„vratiti se iz prekidnog načina“* vraća u procesor sadržaj programskog brojila i prevodi procesor u korisnički način rada.
Gornji slijed instrukcija u opisu funkcija uvijek kada je to moguće skraćeno opisati sljedećom instrukcijom koju možemo vidjeti ispod:

Aktivirati prvu dretvu iz reda pripravne_D;

Tablica 2: Skraćena instrukcija

9.2.2. Funkcije za binarni semafor

Binarni semafor može služiti za ostvarivanje međusobnog isključivanja. „Binarnih semafora može biti više, stoga ih možemo razlikovati po indeksu.“ Za binarni semafor potrebne su nam dvije funkcije:

Ispitati_Bsem(I), koju dretva poziva kada želi ući u kritički odsječak
Postaviti_Bsem(I), koju dretva poziva kada izaz iz kritičnog odsječka

Tablica 3: Funkcije potrebne za binarni semafor

Zaštita kritičnog odsječka ovim funkcijama mogla bi izgledati na idući način:

Ispitati_Bsem(I)
Kritični_odsječak;
Postaviti_Bsem(I);
Nekritični_odsječak;

Tablica 4: Funkcija za zaštitu kritičnog odsječka

Iz opisa djelovanja binarnog semafora slijedi da bi funkcija ispitivanja binarnog semafora trebala izgledati na idući način:

j-funkcija ispitati_bsem(i) {
pohraniti kontekst u opisnik Aktivna_D;
ako je (bsem[i] .v ==1) {
bsem[i] .v ==0;
obnoviti kontekst iz opisnika Aktivna_D;
omogućiti prekidanje;
vratiti se iz prekidnog načina
}
inače {
premjestiti opisnik iz reda Aktivna_D u red bsem[i];
aktivirati prvu dretvu iz reda Pripravne_D;
}
}

Tablica 5: Funkcija ispitivanja binarnog semafora

Navedena funkcija osmišljena je tako da dretva koja naiđe na prolazni semafor nastavlja svoje izvođenje [7].

9.2.3. Funkcije za opći semafor

„Opći semafor služi za sinkronizaciju dretvi i za brojenje događaja i sredstava. Moguće su različite izvedbe općeg semafora. Prvo će se opisati mačin ozvedbe općeg semafora kojim se ostvaruje mehanizam semafora u skladu s opisom E.W. Dijkstre. Taj opći semafor biti će označen kao 0s. Opći semafor se razlikuje od binarnog po tome što će njegova vrijednost biti cijeli broj. Pri kreiranju općeg semafora mora postojati mogućnost da se zada početna vrijednost semafora i koja može biti manja od nule [7].“ Najjednostavnije je promatrati djelovanje takvog semafora onda kada ga ispituje jedna dretva, takav semafor se naziva privatni semafor pridružene mu dretve. Funkcija takvog semafora izgleda ovako:

j-funkcija ispitati_Os(J) {
pohraniti kontekst u opisnik Aktivna_D;
Os[J].v = Os[J].v – 1;
ako je (Os[J].v >= 0) {
obnoviti kontekst iz opisnika Aktivna_D;
omogućiti prekidanje;
vratiti se iz prekidnog načina
}
inače {
premjestiti opisnik iz reda Aktivna_D u red Os[J];
aktivirati prvu dretvu iz reda Pripravne_D;
}
}

Tablica 6: Funkcija općeg semafora

Semafor koji je izveden na način opisan gore može poslužiti kao sinkronizacijski mehanizam u sustavi dretvi.

9.2.4. Funkcije za ostvarivanje kašnjenja

Pri opisivanju strukture podataka jezgre ustanovljeno je na koji način se oblikuje red odgođeniih dretvi. Rečeno je da dretva može zatražiti da sama bude odgođena za M perioda otkucaja sata. U tom slučaju ona će pozvati jezgrinu funkciju Zakasniti_sebe(M).
Ta funkcija će izgledati ovako:

j-funkcija Zakasniti_sebe(M) {
pohraniti kontekst u opisnik Aktivna_D;
uvrstiti opisnik iz reda Aktivna_D u red Odgođene_D;
aktivirati prvu dretvu iz reda Pripravne_D;
}

Tablica 7: Funkcija za ostvarivanje kašnjenja

Isto tako aktivna dretva može još i zatražiti odgođeno izvođenje neke druge dretve s indeksom L i u tom slučaju bi funkcija izgledala na sljedeći način:

j-funkcija Zakasniti (L,M) {
pohraniti kontekst u opisnik Aktivna_D;
premjestiti opisnik iz reda Aktivna_D u red Pripravne_D;
pronaći opisnik dretve L u listi Postojeće_D;
obnoviti kontekst iz opisnika Aktivna_D;
ako je (dretva L nije pasivna) {
dojaviti pogrešku
}
inače {
uvrstiti opisnik dretve L u red Odgođene_D
}
aktivirati prvu dretvu iz reda Pripravne_D;
}

Tablica 8: Funkcija odgođenog izvođenja

Funkcija smije odgođeno pokrenuti neku dretvu samo ako je ona pasivna. Zbog toga se pokušaj pokretanja dretve koja nije bila pasivna mora smatrati pogreškom pa se to posebno dojavljuje.(PARAFEAZE)

9.2.5. Funkcije za ulazne i izlazne operacije

Dretva koja želi obaviti neku ulaznu ili izlaznu operaciju ne može to obaviti neposredno. Ona mora pozvati odgovarajuću funkciju jezgre. U pozivu te funkije ona će specifirati vrstu operacije i adrese izlaznih podataka, odnosno adrese na koje treba smjestiti ulazne podatke. Prema tome, poziv jezgrine funkcije kojim započinje neka ulazno-izlazna operacija razultierati ce sejedecom aktivnošču jezgre

j-funkcija Započeti_UI (K) {
pohraniti kontekst u opisnik Aktivna_D;
premjestiti opisnik iz reda Aktivna_D u red Pripravne_D;
pokrenuti ulazno-izlaznu operaciju na napravi K;
aktivirati prvu dretvu iz reda Pripravne_D;
}

Tablica 9: Funkcija za ulazne i izlazne operacije

10. Datotečni sustav

U ovom poglavlju reći ćemo nešto o datotečnim sustavima modernih operacijskih sustava.

10.1. Uloga datoteka u računalnim sustavima

Sve vrste sadržaja koji se trebaju trajno sačuvati na računalo pohranjuju se u datoteke. Sve stvari koje izvodimo na računalu, oslanjamo se na uporabi i pretvorbi datoteka. Svaki korisnik se u svom prvom pristupu računalu susretne sa datotekama i taj dio operacijskog sustava se naziva datotečni sustav. Sve vrste datoteka čuvaju se na vanjskim spremnicima kao što su: hard disk, ssd ili danas noviji NVMe ssd diskovi. Na svaki od diskova se osim samih datoteka pohranjuju i tablice ili direktoriji u koje se smještaju imena datoteka sa kazaljkama koje omogućuju pristup do datoteka. Datoteke uz svoj naziv dobivaju i sufiks (odvojen je od imena datoteke točkom) kojim se opisuje kojeg je datoteka tipa. Ispod ćemo navesti za najpopularnije datoteke njihove sufikse:

- Programi koji su u strojnom obliku za izvođenje se dodaje sufiks .exe.
- Programi napisani u nekom programskom jeziku dodaje se sufiks .c, .css, .cpp i drugi.
- Multimedijske datoteke (Video(.avi, .mov, .mp4), Glazba(.mp3), Slike(.png, .jpg, .jpeg)

Datotečni sustav provodi različite operacije nad različitim vrstama datoteka. Na primjer datoteka koja drži program za izvođenje može obaviti samo operaciju pokretanja programa. Znakovna datoteka se ne može pokrenuti kao program. Neki tipovi datoteka dopuštaju samo čitanje i pisanje [7].

10.2. Struktura datoteka

Datoteke po strukturi možemo podijeliti na binarne i na nestrukturirane datoteke.

10.2.1. Binarna datoteka

Je najjednostavnija vrsta datoteke koja se koristi za pohranjivanje strojnih programa pripravnih za izvođenje. Datoteka koja je binarna može se promatrati kao niz bitova koji su pohranjeni u bajtove, te se kao cjelina prenose iz jednog spremnika u drugi bez ikakvih promjena.

10.2.2. Nestrukturirana datoteka

U današnjim operacijskim sustavima pretežno se upotrebljavaju nestrukturirane datoteke koje se sastoje od niza bajtova i nemaju strukturu. Datotečni sustav omogućava pristup do pojedinog bajta s pomoću datotaljke (engl. *file pointer*). Prilikom otvaranja nestrukturirane datoteke, datotečna kazaljka postavlja se na početak datoteke. Datotečni sustav omogućuje čitanje niza bajtova iz datoteke i pisanje bajtova u datoteku. Datotečna se kazaljka pri čitanju i pisanju pomiče za broj pročitanih, odnosno upisanih bajtova. Kraj datoteke se obilježava oznakom (engl. *end of file*). Pokušaji čitanja nakon kraja datoteke izbaciti će pogrešku, dok pisanje u datoteku može izazvati produljenje datoteke.

10.3. Smještanje datoteka na disk

Svaka datoteka koja je pohranjena na disk je opisana u datotečnom sustavu svojim opisnikom. Opisnik sadrži sve atribute datoteke. Sadržaji opisnika su:

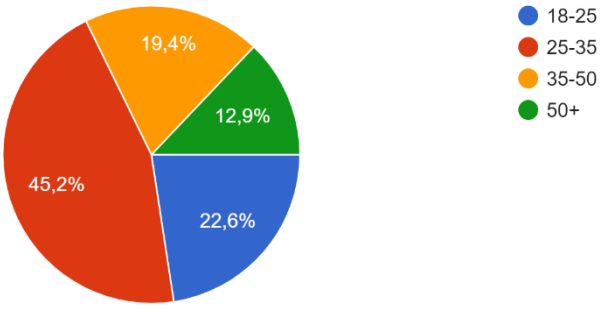
- Naziv datoteke
- Tip datoteke
- Lozinka
- Ime vlasnika datoteke
- Prava pristupa
- Vrijeme stvaranja datoteke
- Vrijeme zadnje uporabe datoteke
- Ime posljednjeg korisnika
- Opis smještaja datoteke na disku

Naziv svake datoteke mora biti jedinstven kako bi je se moglo prepoznati i ne zamijeniti s nekom drugom datotekom. Vlasnik neke datoteke koji zna lozinku može mijenjati atribute datoteke. Datoteke mogu koristiti i drugi korisnici ali samo na način na koji ih je propisao vlasnik datoteke. Najvažnija stvar jest da se datoteka ima gdje spremiti a to je hard disk, SSD ili NVMe SSD [7].

11. Anketa

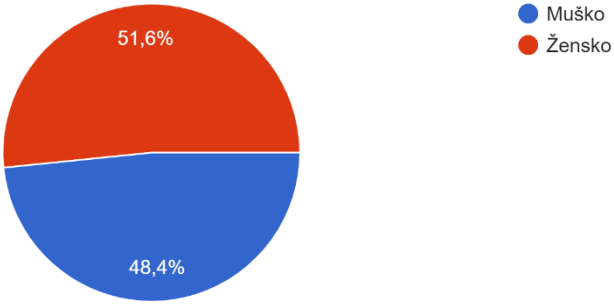
1. Koliko imate godina?

31 odgovor



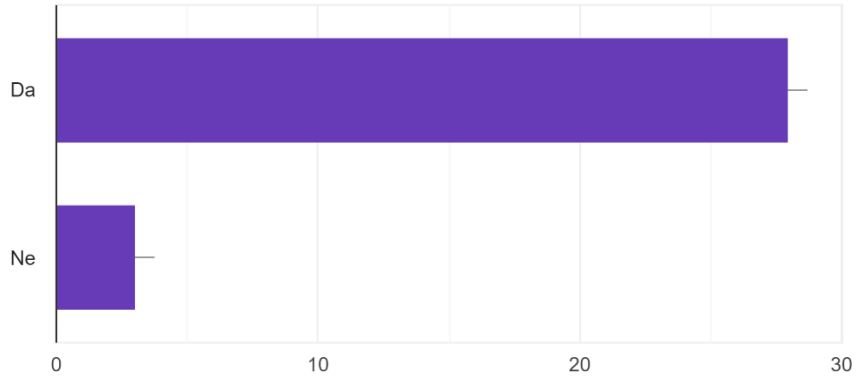
2. Kojeg ste spola?

31 odgovor



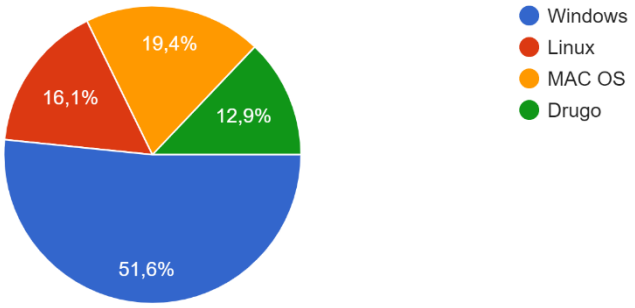
3. Koristite li računalo?

31 odgovor



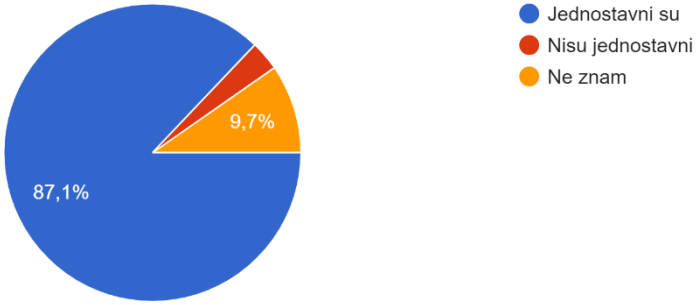
4. Koji operacijski sustav koristite?

31 odgovor



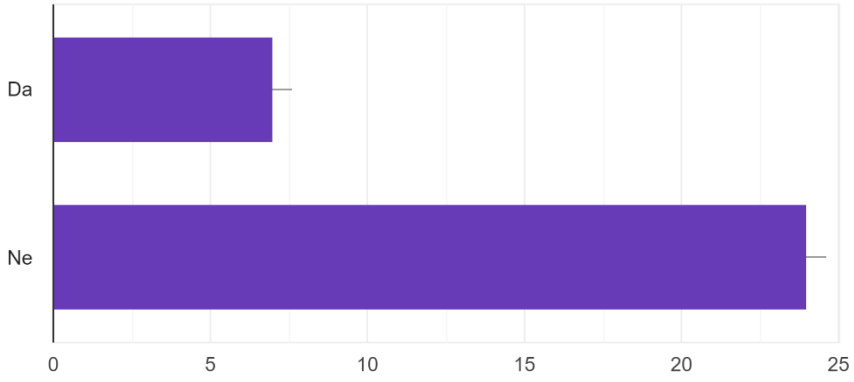
5. Što mislite o operacijskim sustavima?

31 odgovor



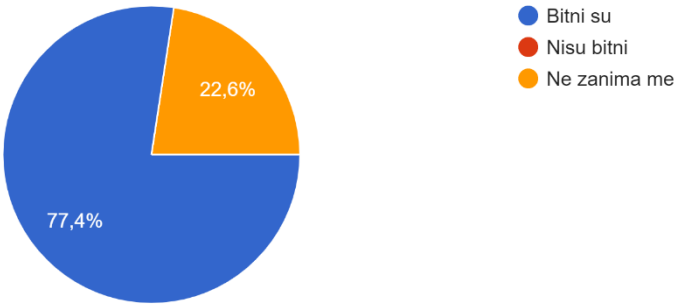
6. Razmišljate li ikada na koji način funkcioniraju operacijski sustavi?

31 odgovor



7. Smatrate li da su operacijski sustavi bitni u današnje vrijeme?

31 odgovor



12. Zaključak

U ovom diplomskom radu detaljno je objašnjen pojam operacijskih sustava kao i njegove temeljne funkcije. „*Operacijski sustavi predstavljaju skup programa koji omogućuju efikasno upravljanje računalnim hardverom, pružaju podršku za korištenje programa te djeluju kao posrednik između korisnika računala i samog računalnog hardvera[7].“* Analizirajući funkcije operacijskih sustava, može se zaključiti kako oni imaju ključnu ulogu u funkcioniranju svih računalnih sustava budući da bez operacijskih sustava, računalo ne može obavljati svoju funkciju. U diplomskom radu prvo se je razradila podjela operacijskih sustava, nakon toga razvoj operacijskih sustava kroz povijest, dotaklo se samih zadataka operacijskih sustava, Von Neummanovog modela računala, naveli su se i opisali ulazno-izlazni uređaji, programi, procesi i dretve, jezgre operacijskog sustava i na samom kraju datotečni podsustav.

Dakle u današnjem svijetu razvijene tehnologije, uloga operacijskih sustava postaje sve značajnija. Glavna zadaća operacijskih sustava prvenstveno je optimizacija korištenja hardvera, a to uključuje upravljanje memorijom, izvršavanje zadaće, upravljanje diskom, grafički prikaz na zaslonu te čitanje i pisanje ulazno-izlaznih jedinica. Moderni operacijski sustavi danas korisnicima omogućuju upravljanje računalom uporabom miša i tipkovnice te obavljanje većeg broja programa istovremeno, jednostavnost korištenja, pouzdanost i stabilnost. S obzirom da se operacijski sustavi konstantno ažuriraju, nitko ne može jasno predvidjeti što će budućnost donijeti. Želje korisnika se mijenjaju na dnevnoj bazi, a samo oni operacijski sustavi koji se u potpunosti prilagode potrebama krajnjeg korisnika biti će uspješni.

12. Literatura

[1] Ivanjek V. (2010) *Povijest operacijskih sustava*, Pristupano 30.5.2021.

http://www.zemris.fer.hr/~leonardo/os/dodatno/Seminar-Povijest-OS-a.pdf

[2] Wikipedia (21.03.2021.) *Operacijski sustav*, Pristupano 30.05.2021.

https://hr.wikipedia.org/wiki/Operacijski_sustav

[3] Theverge.com (2021.) *Windows 8 review,* Pristupano 31.05.2021.

https://www.theverge.com/2012/10/23/3537710/windows-8-review

[4] Linuxzasve.com (2012.) *Kako je nastao Linux*, Pristupano 2.06.2021.

https://www.linuxzasve.com/kako-je-nastao-linux

[5] Wikipedia (2021.) *Windows Vista*, Pristupano 2.06.2021.

https://hr.wikipedia.org/wiki/Windows_Vista

[6] Tvrtko Aščić (2020.) *Linux i njegove distribucije*, Pristupano 2.06.2021.

https://zir.nsk.hr/islandora/object/vus%3A1708/datastream/PDF/view

[7] Budin L., Golub M., Jakobović D., Jelenković L. : *Operacijski sustavi, 1. Izdanje*, Zagreb 2010.

[8] Nino Dasko (2015.) Operacijski sustavi, Pristupano 30.5.2021

https://zir.nsk.hr/islandora/object/unipu%3A184/datastream/PDF/view

[9] Doris Ćuća-Žentil (2018.) *Usporedba operacijskih sustava za mobilne uređaje*, Pristupano 25.6.2021.

https://repozitorij.veleri.hr/islandora/object/veleri%3A1563/datastream/PDF/view

[10] Ante Pavlović (2018) *UBUNTU Linux operacijski sustav*, Pristupano 25.7.2021.

https://zir.nsk.hr/islandora/object/infri:293

[11] Leonardo Jelenković (2014) *Materijali za predavanja: Operacijski sustavi*, Pristupano 4.06.2021.

http://www.zemris.fer.hr/~leonardo/os/dodatno/OS_concepts/

[12] Weebly (bez dat.) *Operacijski sustavi*, Pristupano 29.7.2021.

https://operacijskisustavi.weebly.com/podjela-os-a.html

13. Popis Slika

Slika 1: Prikaz modernog Mac OS-a 13
Slika 2: Prva verzija Windowsa (Windows 1.0) 14
Slika 3: Izgled druge verzije Windowsa (Windows 2.0) 15
Slika 4: Microsoft Windows 3.0 16
Slika 5: Windows NT 17
Slika 6: Prikaz Windowsa 95 18
Slika 7: Windows 98 19
Slika 8: Windows 2000 20
Slika 9: Windows XP 21
Slika 10: Windows Vista 2006 22
Slika 11: Windows 7 23
Slika 12: Windows 8 24
Slika 13: Specifikacije potrebne za rad Windows-a 10 na računalu 25
Slika 14: Windows 10 25
Slika 15: Linus Benedict Torvalds 27
Slika 16: Von Neummanov model računala 29
Slika 17: Prikaz sabirničkog povezivanja dijelova računala 30
Slika 18: Prikaz višedretvenog rada na višeprocesorskom računalu 31
Slika 19: Prikaz spajanja ulazno-izlaznih naprava na sabirnicu računala 34
Slika 20: Shematski prikaz pristupnog sklopa za prijenos pojedinačnih znakova 35
Slika 21: Shematski prikaz ulaza pojedinačnog znaka 36
Slika 22: Prikaz protokola dvožičnog rukovanja 37
Slika 23: Shema jednostavnog modela za neposredni pristup spremniku 38
Slika 24: Programski kod pristupnog sklopa 39
Slika 25: Shema prepuštanja spremničkog ciklusa pristupu 39
Slika 26: Prikaz rada priručnih procesora 40
Slika 27: Prikaz modela višedretvenosti 43
Slika 28: Dretva, domena i kodomena 43
Slika 29: Grafički prikaz podzadatka 45
Slika 30: Cjevovodno ponašanje trodretvenog zadatka 47
Slika 31: Ciklične dretve u ostvarenju cjevovodne obrade 47
Slika 32: Slikovni prikaz jednostavnog modela jezgre 49
Slika 33: Sadržaj opisnika dretve 51

14. Popis Tablica

Tablica 1: Prikaz instrukcije za pohranjivanje konteksta 53
Tablica 2: Skraćena instrukcija.....53
Tablica 3: Funkcije potrebne za binarni semafor.....54
Tablica 4: Funkcija za zaštitu kritičnog odsječka.....54
Tablica 5: Funkcija ispitivanja binarnog semafora.....54
Tablica 6: Funkcija općeg semafora.....55
Tablica 7: Funkcija za ostvarivanje kašnjenja 56
Tablica 8: Funkcija odgođenog izvođenja 56
Tablica 9: Funkcija za ulazne i izlazne operacije 57

Sveučilište
Sjever

SVEUČILIŠTE
SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, IVAN KLANAC (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom OPERACIJSKI SUSTAVI NA RAČUNALIMA (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)
IVAN KLANAC
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, IVAN KLANAC (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom OPERACIJSKI SUSTAVI NA RAČUNALIMA (upisati naslov) čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)
IVAN KLANAC
(vlastoručni potpis)