ARHITEKTURA SUVREMENIH OPERACIJSKIH SUSTAVA

Palić, Martina

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Preddiplomski studij (Poslovna informatika)

Martina Palić

ARHITEKTURA SUVREMENIH OPERACIJSKIH SUSTAVA

Završni rad

Osijek, 2021.

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Preddiplomski studij (Poslovna informatika)

Martina Palić

ARHITEKTURA SUVREMENIH OPERACIJSKIH SUSTAVA

Završni rad

Kolegij: Informatika

JMBAG: 0010226603

e-mail: mapalic@efos.hr

Mentor: (prof. dr. sc. Josip Mesarić)

Osijek, 2021.

Josip Juraj Strossmayer University of Osijek

Faculty of Economics in Osijek

Undergraduate Study (Business Informatics)

Martina Palić

ARCHITECTURE OF MODERN OPERATING SYSTEMS

Final paper

Osijek, 2021.

IZJAVA

O AKADEMSKOJ ČESTITOSTI, PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA, SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je završni rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska.* (cc) BY-NC-SA
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

Ime i prezime studenta/studentice: Martina Palić

JMBAG: 0010226603

OIB: 79191341596

e-mail za kontakt: martina17palic@gmail.com

Naziv studija: Preddiplomski sveučilišni studij, smjer Poslovna informatika

Naslov rada: Arhitektura suvremenih operacijskih sustava

Mentor završnog rada: prof. dr. sc. Josip Mesarić

U Osijeku, 9. rujna 2021. godine

Potpis Martina Palić

Arhitektura suvremenih operacijskih sustava

SAŽETAK

Operacijski sustav predstavlja programe koji obavljaju operacije računala i bez kojih računalni sustav ne bi mogao funkcionirati. Osim toga, omogućuje izvođenje određenih zadataka koje zatraže korisnici, pokreće sve dijelove računala i nadzire njihov rad. Neke od glavnih uloga operacijskog sustava su: olakšavanje primjene računala, učinkovito korištenje svih dijelova računala, omogućavanje višeprogramskog rada i interakcije između više računala povezanih u mreži te upravljanje računalnim resursima koji su važni za rješavanje pojedinačnih problema na djelotvoran način. Većina operacijskih sustava ima nekoliko osnovnih koncepata na kojima su izgrađeni, a u ovom radu ukratko su objašnjeni procesi, adresni prostori, datoteke, ulaz/izlaz i sigurnost. Svrha rada je provesti analizu arhitekture suvremenih operacijskih sustava pa su opisane glavne vrste arhitektura. Operacijski sustav mora se pažljivo kreirati kako bi ispravno radio, a dijeli se na komponente koje bi trebale biti dobro definirani dio sustava i koje čine arhitekturu operacijskog sustava. U ovom radu dan je primjer arhitekture operacijskog sustava Microsoft Windows. Ovaj operacijski sustav odvaja aplikacije od samog operacijskog sustava. Kod jezgre operacijskog sustava radi u jezgrenom načinu rada i ima pristup hardveru i sistemskim podacima, dok aplikacijski kod radi u korisničkom načinu rada i ima ograničen skup dostupnih sučelja, ograničen pristup podacima sustava te nema izravan pristup hardveru. Postoje brojne prednosti, a tako i nedostaci operacijskih sustava, a kako bi se nedostaci smanjili potrebno je izgraditi bolje, sigurnije i pouzdanije operacijske sustave koji će učiniti pomak u odnosu na sadašnje operacijske sustave.

Ključne riječi: operacijski sustav, arhitektura, Windows, korisnički način rada, jezgreni način rada

Architecure of modern operating systems

ABSTRACT

The operating system represents programs that perform computer operations and without which the computer system would not be able to function. In addition, it allows performance of specific tasks requested by users, run all parts of the computer, and supervises their work. Some of the main roles of the operating system are: facilitating the use of computers, efficient use of all parts of the computer, enabling multi-program operation and interaction between multiple computers connected in a network, and managing computing resources that are important to solve individual problems effectively. Most operating systems have several basic concepts on which they are built, and this paper briefly explains processes, address spaces, files, input/output, and security. The purpose of this paper is to analyze the architecture of modern operating systems, so the main types of architectures are described. The operating system must be carefully designed to work properly, and is divided into components that should be a well-defined part of the system and that make up the operating system architecture. This paper gives an example of the architecture of the Microsoft Windows operating system. This operating system separates applications from the operating system itself. The operating system kernel code runs in kernel mode and has access to hardware and system data, while the application code runs in user mode and has a limited set of available interfaces, limited access to system data, and no direct access to hardware. There are numerous advantages, and thus disadvantages of operating systems, and in order to reduce the disadvantages it is necessary to build better, safer and more reliable operating systems that will make a shift compared to current operating systems.

Keywords: operating system, architecture, Windows, user mode, kernel mode

SADRŽAJ

1. Uvod

Operacijski sustavi važan su dio svakog računalnog sustava, a predstavljaju grupu osnovnih programa koji omogućuju obavljanje operacija računala. Također, omogućuju što prikladniju uporabu računala te što djelotvornije iskorištavanje komponenti računalnog sustava. Operacijski sustav korisniku pruža okruženje u kojem može izvršavati zadatke na prikladan i učinkovit način. Ako korisnik želi koristiti računalo za neke jednostavne zadatke, ne mora znati puno više od korisničkog sučelja s kojim se upoznaje prvi prvim susretom s računalom. Međutim, oni korisnici koji su zahtjevniji ili koji žele održavati i modificirati programe moraju detaljnije poznavati osnovne koncepte i komponente na kojima se temelji operacijski sustav. Operacijski sustavi su organizirani na puno različitih načina pa se mogu dosta razlikovati u sastavu. Suvremeni operacijski sustav je velik i složen sustav koji se mora pažljivo kreirati kako bi ispravno radio i lako se mogao doraditi. Sustav se obično dijeli na manje komponente ili module. Svaka od tih komponenti bi trebala biti dobro definirani dio sustava, s pažljivo definiranim funkcijama i sučeljima. S razvojem arhitekture računalnog sustava, razvija se i arhitektura operacijskog sustava i obrnuto.

Svrha ovog rada je na temelju dostupne literature provesti analizu arhitekture suvremenih operacijskih sustava za osobna i prijenosna računala. Osim toga, ovaj rad donosi kratak opis operacijskih sustava, njihovih uloga, glavnih koncepata te vrsta. Završni rad se sastoji od osam poglavlja u kojima je ukratko objašnjeno funkcioniranje suvremenih operacijskih sustava s naglaskom na njihovu arhitekturu. U uvodnom poglavlju definirana je svrha rada te se uvodi čitatelja u temu. U drugom je poglavlju pojašnjena metodologija rada koja obuhvaća cilj istraživanja, korištene metode te izvore podataka. Treće poglavlje donosi uvod u same operacijske sustave s glavnim ulogama, zadacima i navedenim te objašnjenim vrstama. Četvrto poglavlje obuhvaća glavne koncepte operacijskih sustava: procese, adresne prostore, datoteke, ulaz/izlaz i sigurnost. U petom su poglavlju obrađene glavne arhitekture suvremenih operacijskih sustava, a iduće donosi primjer arhitekture Microsoft Windows operacijskog sustava s navedenim komponentama koje se nalaze u korisničkom i jezgrenom načinu rada i formiraju samu arhitekturu. Sedmo i osmo poglavlje odnosi se na raspravu i zaključak u kojima je dano zaključno razmatranje o temi završnog rada.

2. Metodologija rada

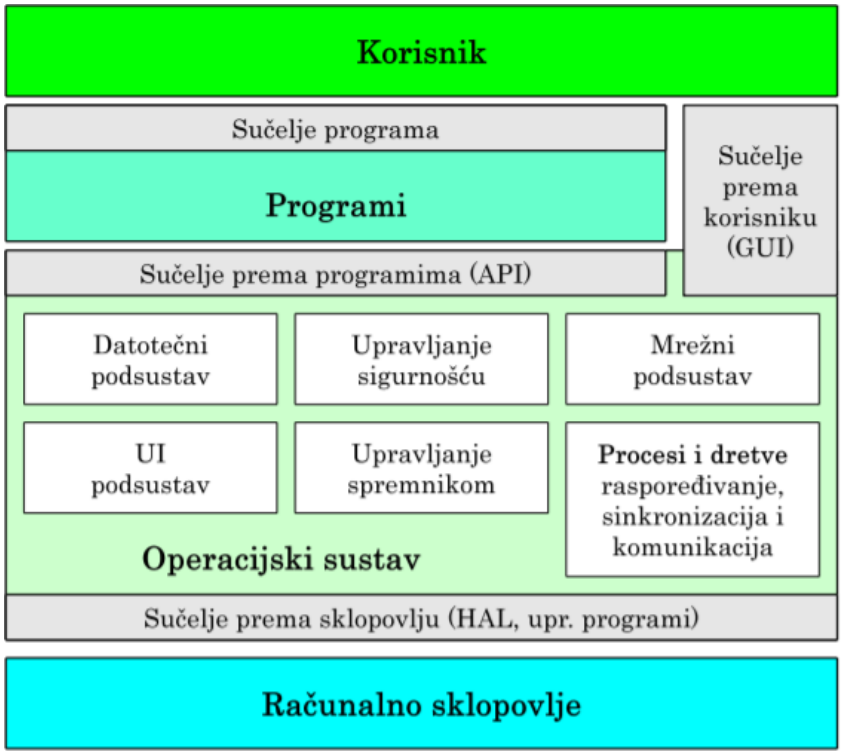
U metodologiji rada pojašnjava se cilj istraživanja završnog rada, iznose se korištene metode istraživanja te se navode izvori podataka i informacija na kojima je temeljen završni rad. Cilj ovog završnog rada je analiza arhitekture suvremenih operacijskih sustava za osobna i prijenosna računala. U kontekstu glavnih funkcija operacijskog sustava objašnjene su uloge i koncepti operacijskih sustava te glavne vrste arhitektura s naglaskom na arhitekturu Microsoft Windows operacijskog sustava.

Za izradu završnog rada korištene su osnovne metode istraživanja poput metode deskripcije kojom se vrši postupak opisivanja određenih pojmova, činjenica i teorije. Metoda klasifikacije je korištena prilikom raznih podjela unutar operacijskih sustava te same podjele arhitektura. Sljedeća korištena metoda je metoda analize kojom se analiziraju komponente korisničkog i jezgrenog načina rada u arhitekturi operacijskog sustava. Korištena je i metoda kompilacije koja se provodi na temelju proučavanja literature drugih autora. Osim navedenih, još je korištena metoda komparacije kojom su opisane razlike u arhitekturama operacijskih sustava.

Za istraživanje teme arhitekture suvremenih operacijskih sustava korišteni su sekundarni izvori podataka iz relevantnih članaka, prezentacija i knjiga koji su javno dostupni i objavljeni na internetu, a pokrivaju područje operacijskih sustava i njihove arhitekture.

3. Operacijski sustavi

Računalni sustav je sačinjen od procesora, radnog spremnika, vanjskih spremnika te raznih ulazno/izlaznih uređaja. Ti elementi zapravo predstavljaju sklopovlje računala. Kako bi sklopovlje računala bilo korisno, uz njega bi trebala funkcionirati adekvatna programska oprema. Razni primjenski programi pretvaraju računalni sustav u određeni virtualni stroj. Za podršku svim tim mnogobrojnim programima koristi se grupa osnovnih programa koji omogućuju obavljanje operacija računala. Ta grupa programa se naziva operacijski sustav. Operacijski sustav omogućuje izvođenje određenih zadataka koje zatraže korisnici, pokreće sve dijelove računala i nadzire njihov rad (Budin i dr., 2010).



Slika 1. Računalni sustav, podsustavi OS-a

Izvor: http://www.zemris.fer.hr/~leonardo/os/fer/_OS-skripta.pdf (preuzeto 10. lipnja 2021.)

Slika 1. prikazuje komponente od kojih se sastoji računalni sustav te podsustave operacijskog sustava. Arhitektura suvremenih operacijskih sustava većinom prati slojeve prikazane na slici. Slojevi su obično podijeljeni na podslojeve jer su operacijski sustavi znatno zahtjevniji (Jelenković, 2020).

Između slojeva sustava se pojavljuje sučelje. Ono regulira način upotrebe i interakcije s korisnicima. Sučelje određuje način na koji se mogu pokretati operacije koje osigurava operacijski sustav. Kako bi se operacije mogle pokrenuti potrebno je postaviti određene zahtjeve. Zahtjeve može postaviti korisnik pomoću korisničkog sučelja ili program pomoću sučelja primjenskog programa. Pomoću navedenih sučelja operacijski sustav vraća ishode traženih operacija ili dobiva informacije o izvršenom zahtjevu. Način postavljanja određenih zahtjeva za provođenje operacija treba biti dogovoren kako bi se uspostavila komunikacija između odvojenih cjelina (Budin i dr., 2010).

3.1. Uloga i zadaci operacijskih sustava

Jedna od glavnih uloga operacijskog sustava je olakšavanje primjene računala. Operacijski sustav omogućuje prikrivanje određenih elemenata i detalja koji korisnicima nisu važni prilikom izvršavanja zadataka koje zahtijevaju. U okviru računala se može u isto vrijeme odigravati više zadataka. Prema tome, bitna uloga operacijskog sustava je učinkovito korištenje svih dijelova računala odnosno strojnih elemenata i programa. Operacijski sustav mora podržati višeprogramski rad te treba osigurati procesoru prebacivanje izvršavanja s jednog niza naredbi na drugi niz. Svaki pojedinačni program treba imati pristup prema nužnim datotekama i ostalim potrebnim sredstvima koje im omogućuje operacijski sustav. Kako bi potrebna sredstva bila što bolje iskorištena, operacijski sustav ih treba dodijeliti određenim programima te ih oduzimati drugim programima. Također, operacijski sustav treba omogućiti interakciju između više računala povezanih u mreži. Osim toga, operacijski sustav raspoređuje i upravlja računalnim resursima koji su neophodni za rješavanje pojedinačnih problema na djelotvoran način (Budin i dr., 2010).

Postoji nekoliko važnih zadataka koje operacijski sustav obavlja (Mesarić, 2015):

- upravljanje procesima
- upravljanje radnom memorijom
- upravljanje sekundarnom memorijom
- upravljanje ulazom/izlazom
- upravljanje datotekama
- zaštita dijelova sustava
- otkrivanje pogrešaka u radu sustava

- korištenje sučelja i upravljačkih naredbi
- umrežavanje.

3.2. Vrste operacijskih sustava

Operacijski sustavi su se razvijali s vremenom pa postoji mnogo vrsta, a sljedeće vrste su neke od najvažnijih koje se najčešće upotrebljavaju.

Batch operacijski sustav: neki računalni procesi su dosta složeni i dugotrajni, a kako bi se oni ubrzali potrebno je grupirati zadatke. „U ovom tipu operacijskog sustava korisnik nema direktnu interakciju s računalom, već mora predati zadatak računalnom operateru. Zatim računalni operater grupira određeni broj zadataka i postavlja ih na ulazni uređaj. Nakon toga specifičan program upravlja izvođenjem svakog programa u grupi.“ Batch operacijski sustav je dosta stara vrsta operacijskog sustava i danas se rijetko koristi (prevedeno prema Studytonight, 2021).

Time-sharing operacijski sustav: dijeljenje vremena je metoda koja omogućuje korištenje određenog računalnog sustava u isto vrijeme korisnicima koji se nalaze na različitim lokacijama. Dijeljenje vremena se naziva još i višezadaćnost (eng. multitasking) te je logično proširenje multiprogramiranja. Dijeljenje vremena zapravo predstavlja procesorsko vrijeme koje se istodobno dijeli s više korisnika. Najveća razlika između time-sharing sustava i multiprogramiranih batch sustava je u tome što batch sustavi imaju za cilj povećati korištenje procesora, a cilj sustava za dijeljenje vremena je smanjenje vremena odgovora. Kako bi se svakom korisniku osiguralo vrijeme, operacijski sustav upotrebljava CPU (eng. Central Processing Unit) planiranje na način da svakom korisniku dodjeljuje određeno vrijeme procesora u kojem se izvodi segment njegovog procesa odnosno programa u izvršenju (Tutorialspoint, 2021).

Operacijski sustav u stvarnom vremenu: definiran je kao sustav koji obrađuje podatke čim su uneseni odnosno čim su oni pohranjeni u RAM računala. Prvo se obrađuju najvažniji podaci pa je vrijeme odgovora korisniku u ovom sustavu maksimalno smanjeno. Sustavi u stvarnom vremenu se upotrebljavaju kada vrijede specifični vremenski zahtjevi koji se odnose na protok podataka ili rad procesora, a mogu se rabiti i kao upravljački uređaji u određenoj namjenskoj aplikaciji. Kako operacijski sustav u stvarnom vremenu ne bi podbacio, mora imati fiksna vremenska ograničenja. Postoje dvije vrste sustava u stvarnom vremenu, a to su hard real-time

sustav i soft real-time. Hard real-time sustav omogućuje maksimalno vrijeme za kritične zadatke te ih odrađuje na vrijeme. U soft real-time sustavu kritični zadatak će imati prioritet u odnosu na druge zadatke, ali se ne jamči sigurnost njegovog dovršetka u definiranom vremenu (Tutorialspoint, 2021).

Distribuirani operacijski sustav: u distribuiranom operacijskom sustavu postoje razni sustavi koji imaju vlastiti procesor, resurse i memoriju. Svi ti sustavi imaju zajedničku komunikacijsku mrežu preko koje su međusobno povezani. U ovom operacijskom sustavu svaki sustav može odrađivati svoj zadatak pojedinačno, neovisno o drugim sustavima. Korisnik ima mogućnost pristupa podacima u drugom sustavu te prema tome može obavljati svoje zadatke (Tutorialspoint, 2021).

Mrežni operacijski sustav: ovaj operacijski sustav ima središnji server te nudi mogućnost upravljanja podacima, sigurnosti, grupama, korisnicima, aplikacijama i ostalim mrežnim funkcijama. Mrežni operacijski sustav ima limitirani pristup podacima na računalima koja su priključena u mrežu, a može pristupiti samo računalnom sklopovlju računala. Osnovna zadaća ovog operacijskog sustava je dopustiti zajedničku datoteku i pristup pisaču između većeg broja računala u mreži, uglavnom je to privatna, lokalna ili neka druga mreža. Neki od primjera mrežnih operacijskih sustava su Linux, Unix, Mac OS X, BSD, Microsoft Windows Server 2008 (Tutorialspoint, 2021).

Višeprocesorski operacijski sustav: odnosi se na primjenu više od jedne središnje procesne jedinice (CPU) u okviru jednog računalnog sustava. Ti višestruki procesori dijele memoriju, računalnu sabirnicu i druge uređaje te su u bliskoj komunikaciji. Višeprocesorski operacijski sustav se upotrebljava kada je neophodna izuzetno velika brzina i snaga pri obradi velike količine podataka. Ovaj sustav se zasniva na simetričnom višeprocesorskom modelu u kojem svaki procesor pokreće jednake kopije operacijskog sustava koje međusobno komuniciraju. Višeprocesorski sustav se rabi u okruženjima kao što su vremenske prognoze, satelitske kontrole i slično (Studytonight, 2021).

Handheld operacijski sustav: uključuje mobilne telefone koji se mogu povezati na mrežu kao što je Internet i osobne digitalne asistente (PDAs) kao što su Palm-piloti. Većina ručnih uređaja ima spore procesore, male zaslone i malu količinu memorije jer su ograničene veličine. Pošto su to uređaji s malom memorijom, operacijski sustav i aplikacije bi trebali efikasno upravljati tom memorijom (Studytonight, 2021).

Embedded operacijski sustav: specijalizirani je operacijski sustav osmišljen na način da izvodi određeni zadatak za neki uređaj koji nije računalo. Primarna zadaća ovog operacijskog sustava je aktiviranje koda kako bi omogućio uređaju obavljanje svog zadatka. Embedded operacijski sustav čini da hardver uređaja bude dostupan softveru koji se pokreće u operacijskom sustavu. Taj sustav je zapravo računalo koje podržava stroj, a neki od primjera uređaja koji koriste ovaj operacijski sustav su semafori, dizala, bankomati, računala u automobilima, digitalne televizije, pametna brojila, navigacijski sustavi i slično (AfterAcademy, 2019).

4. Koncepti operacijskih sustava

Prema Tanenbaumu i Bosu (2015) glavni koncepti operacijskog sustava su:

- procesi
- adresni prostori
- datoteke
- ulaz/izlaz
- sigurnost.

Svaki od njih ukratko će biti objašnjen u nastavku.

4.1. Procesi

Proces je najvažniji koncept svih operacijskih sustava. Proces je zapravo program u izvođenju. Vezan uz svaki proces je njegov adresni prostor kojeg proces može čitati i pisati, a označava popis lokacija memorije od nula do nekog maksimuma. Izvršni program, njegovi podaci i stog su sadržani u adresnom prostoru. Stog (eng. stack) predstavlja apstraktnu strukturu podataka gdje je dozvoljeno dodavanje i izbacivanje elemenata na samo jednom kraju liste. Grupa resursa također je povezana sa svakim procesom, što često uključuje registre poput pokazivača stoga i brojača programa, popise povezanih procesa, popis otvorenih datoteka i sve druge informacije potrebne za rad programa. Proces je u osnovi spremnik koji obuhvaća sve potrebne informacije kako bi se program pokrenuo.

Kako bi proces bio što razumljiviji, najjednostavniji način je razmišljanje o sustavu s više programa. Korisnik može pokrenuti više programa istovremeno, a u međuvremenu se može pokrenuti određeni pozadinski postupak tako da postoji nekoliko aktivnih procesa. Ako je prvi proces iskoristio više od svog udjela procesorskog vremena u posljednjih nekoliko sekundi, operacijski sustav može povremeno blokirati izvođenje tog prvog procesa i krenuti izvoditi drugi. Proces koji je bio privremeno zaustavljen, mora se opet pokrenuti u posve istom stanju u kojem je bio ranije. Drugim riječima, sve informacije o procesu se moraju spremiti na neko mjesto tijekom obustave. Te informacije, osim sadržaja vlastitog adresnog prostora, su spremljene u tablicu operacijskog sustava koja se zove procesna tablica, a predstavlja niz struktura koje sadrže trenutno postojeće procese.

Stvaranje i završavanje procesa najvažniji su sistemski pozivi upravljanja procesima. Kao primjer se može uzeti proces koji se naziva interpreter naredbi, često poznat kao ljuska, koji čita naredbe s terminala. Kada korisnik upiše naredbu koja uključuje kompiliranje programa, ljuska mora aktivirati poseban proces kako bi pokrenuo kompajler. Nakon što proces završi kompilaciju, treba izvesti sistemski poziv kako bi se sam završio. Proces može pokrenuti više drugih procesa, a ti drugi procesi mogu stvoriti još podređenih procesa koji se nazivaju child (dijete) procesi. Kako bi se izvršio određeni zadatak, međusobno povezani procesi koji surađuju bi trebali sinkronizirati svoje akcije i uzajamno komunicirati. Ova vrsta komunikacije poznata je kao međuprocesna komunikacija. Postoje i drugi sistemski pozivi procesa koji oslobađaju memoriju koja nije iskorištena ili zahtijevaju dodatnu memoriju i koji prekrivaju svoj program drugim kada dijete proces završi.

Postoje slučajevi kada treba prenijeti informacije u pokrenuti proces koji te informacije ne čeka. Proces, na primjer, u interakciji s drugim procesom na zasebnom računalu to čini slanjem poruka udaljenom procesu putem računalne mreže. Pošiljatelj poruke može zahtijevati od operacijskog sustava obavijest koju će mu poslati nakon određenog broja sekundi kako bi spriječio mogućnost gubljenja poruke i odgovora te kako bi mogao opet, ako potvrda nije stigla, poslati poruku. Operacijski sustav, kada prođe zadani broj sekundi, šalje signal upozorenja koji izaziva proces da na trenutak zaustavi sve što je radio, pohrani svoje registre u stog te krene izvoditi specifične metode obrade signala poput ponovnog slanja vjerojatno izgubljene poruke. Pokrenut proces ponovno se pokreće u stanju u kojem je bio prije signala, kada se obrada signala završi (Tanenbaum, Bos, 2015).

Na temelju prethodnih razmatranja, upravljanje procesima obuhvaća (Mesarić, 2015):

- stvaranje i uklanjanje korisničkih i sistemskih procesa,
- prekidanje i ponovno aktiviranje procesa,
- sinkronizaciju među procesima,
- komunikaciju među procesima,
- razrješavanje potpunog zastoja.

4.2. Adresni prostori

Za svaki program koji se izvršava, svako računalo koristi glavnu memoriju. Samo jedan program može biti u memoriji u jednostavnom operacijskom sustavu. Kako bi se neki drugi

program pokrenuo, prvi se mora ukloniti i na njegovo mjesto u memoriji staviti drugi. Više programa može istovremeno biti u memoriji u naprednijim operacijskim sustavima. U tim operacijskim sustavima je potreban određen mehanizam zaštite da se programi ne bi sputavali kako međusobno tako i s operacijskim sustavom. Mehanizmom zaštite upravlja operacijski sustav iako se mora nalaziti u hardveru.

Prethodno gledište usredotočeno je na kontrolu i zaštitu glavne memorije računala. Podjednako bitno pitanje vezano uz memoriju, drugačije od prethodnih, je upravljanje adresnim prostorom procesa. Kao što je već ranije navedeno u procesima, svaki proces može koristiti određeni niz adresa koje ima, a koje se uobičajeno kreću od nula do nekog maksimuma. Maksimalna količina adresnog prostora nekog procesa manja je od glavne memorije u najjednostavnijem slučaju. Prema navedenom, proces može popuniti svoj adresni prostor, a da u glavnoj memoriji i dalje ima dovoljno mjesta kako bi sve to zadržala.

Brojna računala imaju adrese od 32 ili 64 bita koje stvaraju adresni prostor od 2^{32} ili 2^{64} bajta. Međutim, glavno pitanje je što će biti ako proces ima više adresnog prostora, a želi sve iskoristiti, nego što računalo ima glavne memorije. Kada su se pojavila prva računala, takav postupak nije bio moguć. U današnje vrijeme postoji virtualna memorija koja omogućava operacijskom sustavu da zadrži dio adresnog prostora u glavnoj memoriji, a dio prostora na disku i prebacuje dijelove između njih ako je potrebno. Proces se može pozivati na skup adresa koji stvara operacijski sustav kao apstrakciju adresnog prostora. Adresni prostor može biti ili veći ili manji od fizičke memorije uređaja od koje je odvojen. Upravljanje fizičkom memorijom i adresnim prostorima ključne su stavke onoga što operacijski sustav čini i osigurava (Tanenbaum, Bos, 2015).

4.3. Datoteke

Operacijski sustav zadužen je za sljedeće aktivnosti s datotekama (Vukelić, n.d.):

- stvaranje i uništavanje datoteka,
- stvaranje i uništavanje direktorija,
- osnovne operacije s datotekama,
- fizičko smještanje datoteka na sekundarnu memoriju,
- sigurnosno spremanje datoteka.

Prema Tanenbaumu i Bosu (2015) datotečni sustav je važan koncept kojeg podržavaju praktično svi operacijski sustavi. Jedan od glavnih ciljeva operacijskog sustava je predstaviti programeru čisti apstraktni model datoteka neovisnih o uređaju na način da prikrije posebnosti diskova i drugih ulazno/izlaznih uređaja. Kako bi se datoteka mogla kreirati, ukloniti, čitati i pisati potrebni su sistemski pozivi. Datoteka mora biti postavljena na disk i otvorena da bi se mogla čitati, a kada se pročita treba ju zatvoriti. Upravo za te aktivnosti su zaduženi pozivi. Većina operacijskih sustava osobnih računala ima model direktorija kao način grupiranja datoteka kako bi se omogućilo mjesto na kojem će se one čuvati. Korisnik može imati po jedan direktorij za svaki zadatak koji obavlja i tada su nužni sistemski pozivi za stvaranje i brisanje direktorija. Sistemski pozivi, između ostalog, mogu dodavati postojeću datoteku u direktorij i brisati datoteku iz direktorija. U direktorij se mogu, osim datoteka, unijeti i drugi direktoriji.

Hijerarhije procesa i datoteka strukturirane su kao stabla što je njihova jedina sličnost. Hijerarhije procesa obično nemaju više od tri razine dok je za hijerarhije datoteka uobičajeno da imaju čak pet ili više razina. Hijerarhija direktorija može trajati godinama, a hijerarhija procesa je dosta kratkotrajna, do nekoliko minuta. Osim hijerarhija, procesi i datoteke se razlikuju u vlasništvu i zaštiti. Samo roditelj proces može nadzirati i pristupiti dijete procesu, a kako čitanje datoteka i direktorija ne bi bilo dostupno samo vlasniku, nego i široj grupi, postoje određeni mehanizmi operacijskog sustava kojima se određuje pravo pristupa, čitanja i editiranja.

Operacijski sustav je zadužen za upravljanje datotekama. Dizajn operacijskog sustava bavi se načinom na koje su datoteke strukturirane, imenovane, zaštićene, korištene, kako im se pristupa i upravlja te kako se implementiraju. Svi ti načini kojima se bavi operacijski sustav čine ranije spomenuti datotečni sustav. Najvažnije komponente datotečnog sustava, iz korisničke perspektive, su načini kako se datoteke imenuju i štite, što čini datoteku, kako izgleda stablo direktorija, koje su radnje dopuštene nad datotekama, vrijeme kada su kreirane, ispravljane i pohranjene, veličinu memorijskog prostora kojeg zauzimaju (Tanenbaum, Bos, 2015).

Detalji o tome koliko primjerice ima sektora u logičkom bloku diska nisu toliko zanimljivi korisnicima, ali su iznimno važni za dizajniranje datotečnog sustava. Prema Dmitroviću (2018) sektor predstavlja najmanju veličinu podataka koja se šalje blok uređaju, a korištenjem standarda LBA (eng. Logical Block Addressing) uvodi se pojam logički blok koji „koristi adresiranje od 0 nadalje pa su adrese u jednodimenzionalnom nizu logičkih blokova, gdje je logički blok najmanja jedinica.“ Ovo adresiranje skriva fizičku adresu od operacijskog sustava, a pošto ima više sektora, na vanjskim trakama ima i više podataka. Osim navedenog,

implementatore datotečnog sustava zanima kako se datoteke i direktoriji pohranjuju, kako se upravlja diskovnim prostorom te kako učiniti da sve radi učinkovito i pouzdano.

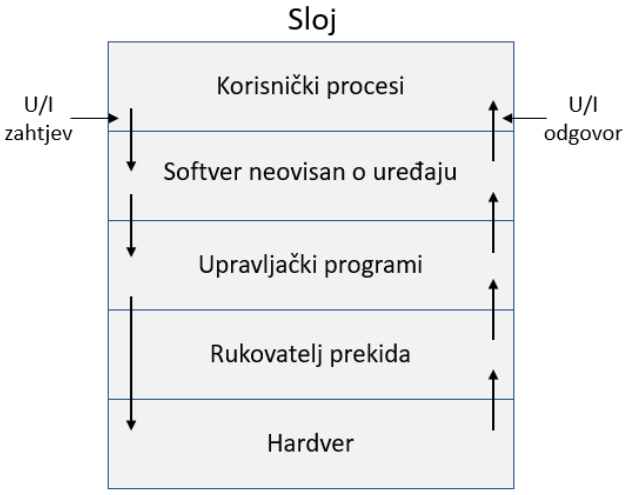
Kao primjer suvremenog datotečnog sustava može se uzeti NTFS (NT File System) sustav koji je razvijen za novije NT verzije Windowsa. NTFS poboljšava funkcionalnost i sigurnost operacijskog sustava Windows, a koristi 64-bitne adrese diska i može podržati particije diska do 2^{64} bajta iako ga druga razmatranja ograničavaju na manje veličine. NTFS omogućuje dohvaćanje i spremanje datoteka na tvrdim diskovima (HDD) i SSD diskovima te se koristi za imenovanje, organiziranje i pohranu datoteka, a osim toga omogućuje i formatiranje SSD diskova, tvrdih diskova, USB-ova i micro SD kartica koje se koriste sa operacijskim sustavom Windows (Tanenbaum, Bos, 2015).

4.4. Ulaz/izlaz

Operacijski sustav, osim što omogućuje upravljanje procesima, datotekama i adresnim prostorima, također upravlja i svim ulazno/izlaznim uređajima računala. Operacijski sustav mora slati naredbe, rješavati pogreške i upravljati prekidima. Osim toga, trebao bi pružiti sučelje između uređaja i ostatka sustava koje će biti jednostavno i lako za korištenje. Sučelje bi za sve uređaje trebalo biti isto, koliko god je to moguće. Ulazno/izlazni kod čini važan dio cjelokupnog operacijskog sustava.

Fizičke uređaje koji omogućuju ulaz i stvaraju izlaz ima svako računalo. Korisnici, uz pomoć tih uređaja, mogu „reći“ računalu što treba raditi i tako dobiti željene rezultate nakon što računalo izvrši traženi zadatak. Sukladno tome, postoji mnogo različitih vrsta ulazno/izlaznih uređaja. Glavni ulazni uređaji svakog računala su tipkovnica i miš, dok su uobičajeni izlazni uređaji monitor i pisač. U skupinu uređaja koji su i ulazni i izlazni mogu se svrstati vanjski spremnici kao što su magnetski diskovi, magnetske diskete i optički diskovi te SSD. Osim navedenih, u današnja računala se mogu priključiti razni ulazno/izlazni uređaji poput mikrofona, zvučnika, videokamere, CD-a, videorekordera, uređaja za čitanje znakova i slika i sl. Nabrojani uređaji su prema svojim fizičkim načelima vrlo različiti, a različita je i brzina rada svakog uređaja. Obično je izvršavanje zadatka u uređajima neovisno o brzini izvođenja programa u procesu pa je prema tome potrebno posvetiti pažnju sinkronizaciji rada procesora i ulazno/izlaznih uređaja (Budin i dr., 2010).

Kao što je već ranije navedeno, operacijski sustav upravlja ulazno/izlaznim uređajima računala pa shodno tome svaki sustav uključuje ulazno/izlazni podsustav za upravljanje svojim ulazno/izlaznim uređajima. Neki ulazno/izlazni programi su neovisni o uređajima, odnosno funkcioniraju na mnogim ili čak svim ulazno/izlaznim uređajima. S druge strane, neki dijelovi programa, kao što su upravljački programi, zaduženi su samo za određene ulazno/izlazne uređaje. Na slici 2. sažeto su prikazani slojevi ulazno/izlaznog sustava. Svaki od slojeva ima određene ulazno/izlazne funkcije koje obavlja.



Slika 2. Slojevi ulazno/izlaznog sustava

Izvor: izrada autora prema Tanenbaum i Bos (2015:368)

Slika prikazuje strelice koje označavaju tok kojim se odvijaju ulazno/izlazni zahtjevi i odgovori. Na primjer, operacijski sustav je pozvan da izvrši poziv kada korisnički program pokuša pročitati blok iz datoteke. Softver neovisan o uređaju potrebni blok traži u priručnoj memoriji, a ako ga nema onda poziva upravljački program kako bi izdao zahtjev hardveru da ga preuzme s diska. Tada se proces zaustavlja sve dok operacija na disku ne bude dovršena i podaci ne budu sigurno dostupni u međuspremniku pozivatelja. Nakon što operacija na disku završi, hardver generira prekid te se pokreće rukovatelj prekida kako bi se otkrilo koji uređaj želi „pažnju“. Rukovatelj prekida onda izvlači status iz uređaja i „budi“ proces kako bi dovršio ulazno/izlazni zahtjev i omogućio nastavak korisničkog procesa (Tanenbaum, Bos, 2015).

4.5. Sigurnost

Sve više korisnika koristi sve više informacija koje se nalaze u raspodijeljenim sustavima, zbog čega postoji sve veća opasnost od uništavanja ili neovlaštene uporabe informacija pa sigurnost računalnih sustava postaje sve bitnija. Kako bi se to spriječilo, postoje mnogi mehanizmi zaštite koji jamče sigurnost računalnih sustava. Na razne načine je moguće razvrstati narušavanje sigurnosti računalnih sustava. Prema Budinu (2010), jedna od podjela sigurnosnih mehanizama je:

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

Mehaničko oštećenje uređaja, oštećenje nastalo požarom ili poplavom, krađa medija i uređaja na kojima su pohranjene informacije ono je što se smatra vanjskim utjecajima. Zaštitne mjere se svode na čuvanje kopija informacija i ograničavanje pristupa prostorima gdje se uređaji nalaze, a pristup je dopušten samo ovlaštenim osobama. Dakle, računalni sustav smiju koristit samo povjerljive osobe. Da bi se utvrdilo pravo pristupa, potrebno je provjeriti identitet korisnika odnosno utvrditi autentičnost identiteta. Kako bi pristupio računalu, korisnik se mora predstaviti procesom identifikacije. Računalni sustav mora provesti autentifikaciju kako bi provjerio identifikaciju korisnika. Ona se vrši prilikom prijave za rad na sustavu, a utvrđuje se autentičnost samo korisnika dok se u raspodijeljenim sustavima često utvrđuje autentičnost i korisnika i računala. Korisnik može upotrebljavati računalna sredstva kada prođe postupak autentifikacije. Ono što korisniku mora omogućiti pristup do određenih sredstava su unutarnji zaštitni mehanizmi. Mehanizmi dopuštanja pristupa nazivaju se autorizacijom pristupa.

Svaki napadač ili uljez može lako ugroziti sigurnost raspodijeljenih sustava jer se informacije prenose nestabilnim komunikacijskim kanalima, a pristup do njih se ne može zaštititi fizički. Zato su komunikacijski zaštitni mehanizmi jedni od važnijih oblika postizanja sigurnosti. Zaštita poruka je glavni problem komunikacijskih zaštitnih mehanizama. Kako bi se poruke zaštitile, najučinkovitije je njihovo kriptiranje. Operacijski sustav, osim za zaštitu poruka, koristi kriptografiju za sigurno spremanje datoteka na disk, za siguran prijenos podataka putem mreže, za kodiranje lozinki u datoteci s lozinkom i slično. Također se koristi i kaljenje programa (eng. hardening) kako bi se osiguralo da svaki proces ima prava koja su mu potrebna za obavljanje posla ili kako bi se onemogućilo napadače da stave novi kod u pokrenuti program.

5. Arhitektura operacijskih sustava

Suvremeni operacijski sustav je velik i složen sustav koji se mora pažljivo kreirati kako bi ispravno radio i lako se mogao doraditi i ažurirati. Umjesto da postoji jedan sustav, obično se dijeli zadatak na manje komponente ili module. Svaka od tih komponenti bi trebala biti dobro definirani dio sustava, s pažljivo definiranim funkcijama i sučeljima. Kako se razvija arhitektura računalnog sustava, tako se razvija i arhitektura operacijskog sustava, a vrijedi i obrnuto. Postoje četiri glavne vrste arhitektura operacijskog sustava (PadaKuu, 2021):

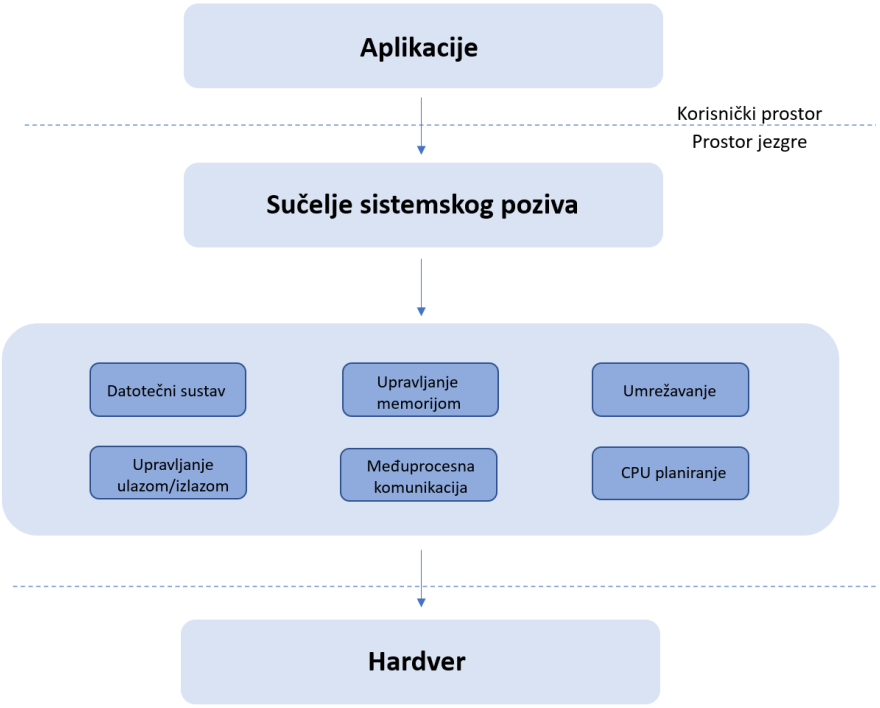
- monolitna arhitektura,
- slojevita arhitektura,
- mikrojezgrena arhitektura,
- hibridna arhitektura.

5.1. Monolitna arhitektura

U monolitnoj arhitekturi, čitav operacijski sustav radi u jezgrenom načinu rada kao jedinstveni program. Jezgra ili kernel predstavlja centralni dio suvremenog operacijskog sustava. Ona pruža razne usluge aplikacijama i upravlja sklopovljem. Osim toga, upravlja računalnim resursima i čini sloj između korisničkih programa i fizičkog računalnog sklopovlja. Operacijski sustav je napisan kao skup postupaka koji su međusobno povezani u jedan veliki izvršni binarni program. Svaki postupak u sustavu može pozvati bilo koji drugi ako taj drugi pruža izračune koji su korisni prvom. Vrlo je učinkovita sposobnost pozivanja bilo kojeg postupka, ali situacija u kojoj postoje stotine postupaka koji se međusobno mogu pozivati bez ograničenja može dovesti do sustava koji je težak i kompliciran. Između ostalog, ako se dogodi da padne bilo koji od postupaka, past će i operacijski sustav. Kada se koristi ovaj pristup, prvo se kompiliraju svi pojedinačni postupci ili datoteke koje sadrže postupke, a nakon toga se pomoću povezivača sustava sve zajedno povezuje u jednu izvršnu datoteku kako bi se kreirao stvarni objektni program operacijskog sustava. Svaki postupak je vidljiv svakom drugom postupku, dakle nema skrivanja informacija (Tanenbaum, Bos, 2015).

Monolitni sustavi mogu imati određenu strukturu. Sistemski pozivi koje pruža operacijski sustav se traže stavljanjem parametara na točno određeno mjesto kao što je stog, a potom izvršavanjem upute za prekid. Ta uputa prenosi kontrolu na operacijski sustav i prebacuje stroj

iz korisničkog načina rada u jezgreni način rada. Nakon toga operacijski sustav određuje koji sistemski poziv treba izvršiti dohvaćajući parametre. U ovom sustavu postoji jedan servisni postupak za svaki sistemski poziv koji ga izvršava. Sve što se nalazi ispod sučelja sistemskog poziva i iznad hardvera čini jezgru. Kao što je prikazano na slici 3., jezgra putem sistemskih poziva pruža razne funkcije operacijskih sustava kao što su upravljanje memorijom, umrežavanje, datotečni sustav, CPU planiranje i dr. Sve funkcije jezgre se izvode u jednom adresnom prostoru (Tanenbaum, Bos, 2015).



Slika 3. Monolitna arhitektura

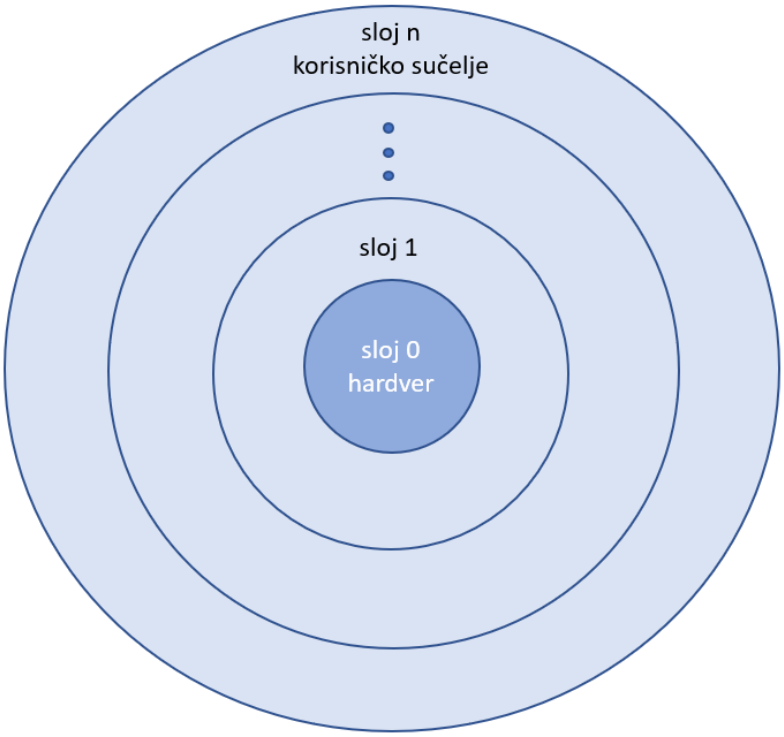
Izvor: izrada autora prema PadaKuu, 2021.

Iako se čini da su monolitne jezgre jednostavne, one su zapravo dosta teške za izvršiti i proširiti. Neke od prednosti monolitnih jezgri su te što u sučelju sistemskog poziva ima vrlo malo dodatnih troškova i komunikacija unutar jezgre je brza. Usprkos nedostacima monolitnih jezgri poput problema sigurnosti, njihova učinkovitost i brzina objašnjavaju zašto i dalje u operacijskim sustavima kao što su Linux i Windows postoje pokazatelji ove strukture (Silberschatz i dr., 2018).

5.2. Slojevita arhitektura

Za razliku od monolitne arhitekture koja je poznata kao čvrsto povezan sustav jer izmjene jednog dijela sustava mogu imati utjecaj na druge dijelove, može se dizajnirati i labavo povezan sustav koji je podijeljen na individualne, manje sastavnice sa specifičnim i ograničenim funkcijama od kojih se sastoji jezgra. Ovaj modularni pristup omogućuje implementatorima sustava veću samostalnost u razvoju i promjeni unutarnjeg rada sustava jer promjene jedne sastavnice ne utječu na promjene druge, već samo na tu sastavnicu što je prednost ovog pristupa.

Postoji više načina kako bi se sustav učinio modularnim, a jedan od njih je slojeviti pristup u kojem je operacijski sustav podijeljen u nekoliko slojeva. Hardver se nalazi u najnižem, donjem sloju dok je korisničko sučelje u najvišem, gornjem sloju kao što je prikazano na slici 4. Implementacija apstraktnog objekta koja se sastoji od podataka i operacija koje mogu manipulirati tim podacima je sloj operacijskog sustava. Klasičan sloj operacijskog sustava se sastoji od skupa funkcija i struktura podataka koje slojevi više razine mogu pozvati. Sloj više razine, s druge strane, može pozvati operacije na slojevima niže razine.



Slika 4. Slojevita arhitektura

Izvor: izrada autora prema Silberschatz i dr. (2018:84)

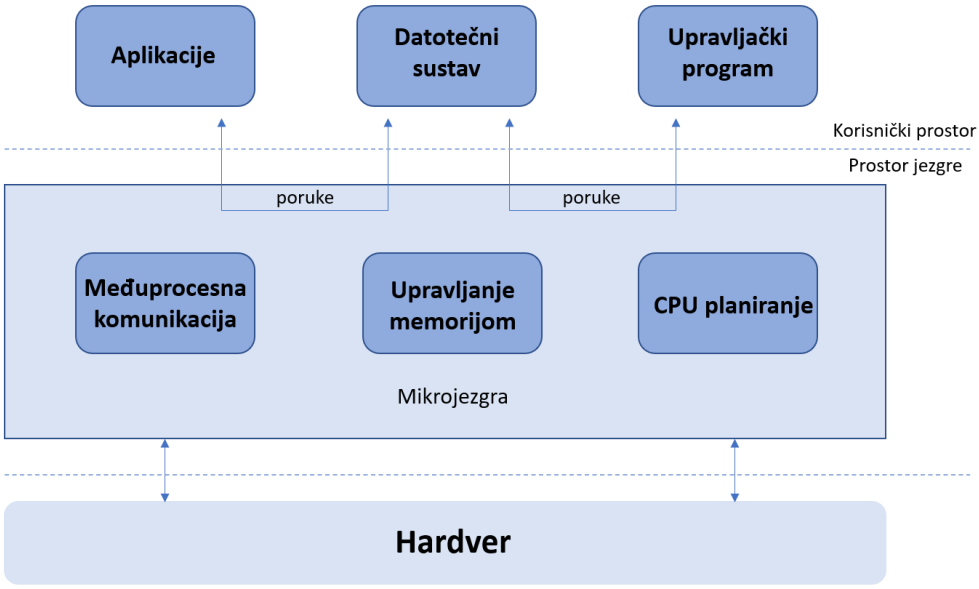
Slojevi su odabrani na način da svaki koristi funkcije i usluge samo slojeva niže razine, prema tome, ovaj pristup olakšava provjeru sustava i otklanjanje pogrešaka, a omogućuje i jednostavnost izrade. Budući da koristi samo osnovni hardver za implementaciju svojih funkcija, prvi se sloj može popraviti bez obzira na ostatak sustava. Jednom kada su ispravljene pogreške prvog sloja, može se pretpostaviti njegovo ispravno funkcioniranje dok se ispravljaju pogreške drugog sloja itd. Ako se tijekom otklanjanja pogrešaka određenog sloja pronađe greška, ona mora biti na tom sloju jer su pogreške na slojevima ispod njega već otklonjene. Kao rezultat toga, dizajn i implementacija sustava su jednostavniji.

Kao što je već ranije spomenuto, svaki sloj je implementiran samo s operacijama koje omogućuju slojevi niže razine. Sloj samo mora znati što rade te operacije, ne i kako se one provode. Shodno tome, svaki sloj skriva, od slojeva više razine, postojanje određenih operacija, struktura podataka i hardvera. Slojeviti sustavi su se pokazali učinkovitima u računalnim mrežama i mrežnim aplikacijama, ali mali broj operacijskih sustava koristi čistu slojevitu arhitekturu. Jedan od razloga su poteškoće u definiranju funkcionalnosti svakog sloja. Ukupna izvedba tih sustava je slaba zbog općih troškova koji zahtijevaju kretanje korisničkog programa kroz više slojeva kako bi dobio uslugu operacijskog sustava. No, neki slojevi su uobičajeni u suvremenim operacijskim sustavima. Ovi sustavi, općenito, imaju manje slojeva s više funkcionalnosti izbjegavajući probleme definiranja i interakcije slojeva uz istodobno pružanje većine prednosti modulariziranog koda (Silberschatz i dr., 2018).

5.3. Mikrojezgrena arhitektura

Kako su se operacijski sustavi širili i razvijali tako je jezgra postala velika i teška za upravljanje. Sredinom osamdesetih godina prošlog stoljeća razvijen je operacijski sustav Mach koji je modulirao jezgru koristeći mikrojezgreni pristup. Ovaj pristup organizira operacijski sustav na način da uklanja sve nevažne elemente iz jezgre te ih implementira kao programe na korisničkoj razini koji se nalaze u odvojenim adresnim prostorima, a kao rezultat se dobije manja jezgra. Dosta je neslaganja oko toga koje usluge treba implementirati u korisnički prostor, a koje trebaju ostati u prostoru jezgre. Mikrojezgre pružaju minimalno upravljanje procesom i memorijom uz mogućnost komunikacije. Pružanje komunikacije između klijentskog programa i različitih usluga koje se nalaze u korisničkom prostoru glavni je zadatak mikrojezgre. Spomenuta komunikacija se pruža pomoću prosljeđivanja poruka. Za primjer se može uzeti klijentski program koji želi pristupiti datoteci, a koji mora komunicirati s datotečnim

poslužiteljem. Usluga i klijentski program nemaju izravnu interakciju, već razmjenjuju poruke s mikrojezgrom. Slika 5. najbolje ilustrira mikrojezgru i komunikaciju između programa i usluga.



Slika 5. Mikrojezgrena arhitektura

Izvor: izrada autora prema Silberschatz i dr. (2018:85)

Olakšavanje proširenja operacijskog sustava jedna je od prednosti mikrojezgrenog pristupa. Nove usluge ne zahtijevaju izmjenu jezgre jer su sve dodane u korisnički prostor, a kada je potrebno izmijeniti jezgru, izmjene su većinom manje jer se radi o manjoj jezgri. Rezultirajući operacijski sustav lakše se prenosi s jednog dizajna hardvera na drugi. Budući da se većina usluga izvodi kao korisnički proces, a ne kao jezgreni, mikrojezgra osigurava veću sigurnost i pouzdanost. Ako jedna od usluga zakaže to neće utjecati na ostatak operacijskog sustava.

Povećani dodatni troškovi funkcije sustava mogu pogoršati performanse mikrojezgara. Poruke se moraju kopirati između usluga koje se nalaze u odvojenim adresnim prostorima kada dvije usluge na razini korisnika moraju komunicirati. Nadalje, kako bi razmijenio poruke, operacijski sustav će se možda morati prebaciti s jednog procesa na drugi. Općeniti troškovi povezani s kopiranjem i prebacivanjem poruka između procesa najznačajnija su prepreka napretku operacijskih sustava temeljenih na mikro jezgrama (Silberschatz i dr., 2018).

5.4. Hibridna arhitektura

Sve arhitekture do sada spomenute imaju svoje prednosti i nedostatke. Monolitne arhitekture su relativno brze, no vrlo ih je teško proširiti. Slojevita arhitektura pruža učinkovitu podjelu funkcionalnosti, ali je teško upravljati sustavom ako je broj slojeva prevelik. Mikrojezgrena arhitektura vrlo je djelotvorna u izdvajanju osnovnih funkcionalnosti unutar mikrojezgre, međutim one usluge koje nisu dio jezgre nisu pravilno integrirane. Stoga je hibridna arhitektura operacijskih sustava nastala kombinacijom najboljih funkcionalnosti ostalih arhitektura. Hibridna arhitektura sadrži tri sloja (PadaKuu, 2021):

- Sloj apstrakcije hardvera: najniži je sloj koji funkcionira kao sučelje između jezgre i hardvera.
- Sloj mikrojezgre: predstavlja mikrojezgru koja se sastoji od tri osnove funkcije: upravljanje memorijom, planiranje procesora i međuprocesna komunikacija.
- Aplikacijski sloj: nalazi se u korisničkom prostoru i funkcionira kao sučelje između korisničkog prostora i sloja mikrojezgre, a uključuje ostale funkcije poput upravljanja ulazno/izlaznim uređajima, datotečni poslužitelj, otkrivanje pogrešaka i dr.

Mali je broj operacijskih sustava koji imaju jednu, strogo definiranu arhitekturu, nego većinom kombiniraju razne arhitekture što stvara hibridne sustave koji rješavaju probleme sigurnosti, performansi i upotrebljivosti. Za primjer se može uzeti Linux koji ima monolitnu arhitekturu koja pruža učinkovite performanse jer se operacijski sustav nalazi u jednom adresnom prostoru. Osim što je monolitan, Linux je i modularan tako da se nova funkcija može dodati u njegovu jezgru. Windows također uglavnom ima monolitnu arhitekturu zbog performansi, iako ima i značajke mikrojezgrene arhitekture poput pružanja podrške za zasebne podsustave koji se izvode kao procesi u korisničkom načinu rada. Osim toga, Windows operacijski sustavi omogućuju dinamički učitane module jezgre. Dakle, hibridna arhitektura je vrlo korisna i uglavnom se koristi u suvremenim operacijskim sustavima (Silberschatz i dr., 2018).

6. Arhitektura Microsoft Windows operacijskog sustava

Windows je suvremeni operacijski sustav koji se pokreće na potrošačkim osobnim računalima, prijenosnim računalima, telefonima i tabletima, a osim njih radi i na poslovnim stolnim računalima te poslovnim poslužiteljima. Microsoftov sustav igara Xbox i Azure infrastruktura računarstva u oblaku također koriste Windows kao svoj operacijski sustav. Windows 10 je najnovija verzija Microsoftovih operacijskih sustava koji se temelje na njegovom NT kodu. NT kod je zamijenio ranije sustave zasnovane na Windows 95/98 (Silberschatz i dr., 2018). U ovom poglavlju ukratko će se opisati arhitektura sustava te njezine komponente.

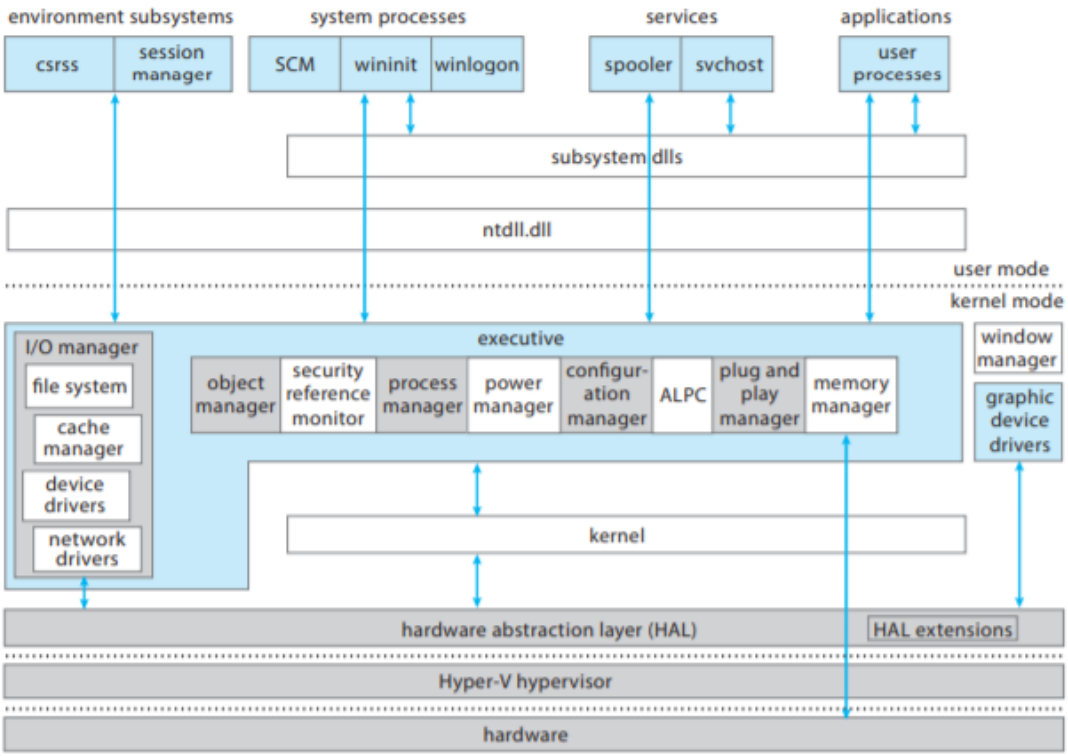
Većina višekorisničkih operacijskih sustava odvaja aplikacije od samog operacijskog sustava. Kod jezgre operacijskog sustava radi u povlaštenom procesorskom načinu rada koji se naziva jezgreni način rada te ima pristup hardveru i sistemskim podacima, za razliku od aplikacijskog koda koji radi u nepovlaštenom procesorskom načinu rada i zove se korisnički način rada s ograničenim skupom dostupnih sučelja, ograničenim pristupom podacima sustava te nema izravan pristup hardveru. Nakon što program u korisničkom načinu rada pozove sistemsku uslugu, procesor izvršava posebnu uputu koja prebacuje nit u jezgreni način rada, a kada se sistemska usluga završi, operacijski sustav prebacuje kontekst niti natrag u korisnički način rada i dopušta pozivatelju da nastavi dalje (Russinovich i dr., 2012).

Znatan dio operacijskog sustava Windows i kod upravljačkog programa dijele isti zaštićeni memorijski prostor koji se nalazi u jezgrenom načinu rada. Prema tome, moguće je da bilo koja komponenta operacijskog sustava ili upravljački program ošteti podatke koje koriste druge komponente operacijskog sustava. Windows pokušava ojačati kvalitetu i ograničiti podrijetlo upravljačkih programa trećih strana kroz razne programe, a istovremeno uključuje i dodatne tehnologije zaštite jezgre kao što je sigurnost temeljena na virtualizaciji. Sve komponente operacijskog sustava su zaštićene od zlonamjernih aplikacija jer aplikacije nemaju izravan pristup kodu i podacima u jezgrenom načinu rada. Ova zaštita je jedan od razloga zašto se Windows smatra snažnim i stabilnim kao aplikacijski poslužitelj i kao platforma radne stanice, a također je i brz u pogledu osnovnih funkcija operacijskog sustava. Komponente jezgrenog načina rada utjelovljuju osnovna objektno-orijentirana načela dizajna, no Windows nije u strogom smislu objektno-orijentirani sustav. Radi prenosivosti, većina koda operacijskog sustava u jezgrenom načinu rada napisana je u C programskom jeziku. Objektno-orijentirane značajke poput polimorfnih funkcija i nasljeđivanja klasa nisu izvorno podržane od navedenog programskog jezika. Shodno tome, implementacija objekata temeljena na C programskom

jeziku u sustavu Windows posuđuje se od značajki pojedinih objektno-orijentiranih jezika, ali ne ovisi o njima (Russinovich i dr., 2012).

Microsoft Windows uključuje široku mrežnu podršku koja je integrirana s aplikacijskim programskim sučeljima (API) Windowsa i ulazno/izlaznim sustavom te je umrežavanje vrlo važna komponenta operacijskog sustava. Protokoli, usluge, aplikacijska programska sučelja i upravljački programi za mrežne adaptere četiri su glavne vrste mrežnih softverskih komponenti, pri čemu svaka komponenta ima sloj na vrhu sljedeće kako bi tvorile mrežni stog. Budući da Windows ima dobro definirana sučelja za svaki sloj, treće strane mogu razvojem vlastitih komponenti, proširiti mrežne mogućnosti operacijskog sustava (Russinovich i dr., 2012).

U nastavku slijedi prikaz arhitekture sustava Windows sastavljene od komponenti koje imaju ključnu ulogu u funkcioniranju operacijskog sustava.



Slika 6. Windows arhitektura

Izvor:

http://www.nastooh.com/teaching/Silberschatz_Operating_System_Concepts_10e_2018.pdf

(preuzeto: 26. lipnja 2021.)

Na slici 6. se vidi isprekidana linija koja dijeli korisnički način rada i jezgreni način rada operacijskog sustava Windows. Okviri iznad isprekidane linije prikazuju procese u korisničkom načinu rada, a komponente ispod linije predstavljaju usluge jezgrenog načina rada operacijskog sustava. Niti korisničkog načina rada izvršavaju se u privatnom adresnom prostoru procesa, a imaju pristup sistemskom prostoru dok se izvršavaju u jezgrenom načinu rada. Kao rezultat toga, podsustavi okoline, sistemski procesi, uslužni procesi i korisnički procesi imaju svaki svoj privatni adresni prostor procesa. Također je vidljiva i druga isprekidana linija koja odvaja dijelove jezgrenog načina rada i hipervizora. Hipervizor je najniži softverski sloj koji se u Windowsu naziva Hyper-V. Svaki operacijski sustav radi u svom virtualnom stroju, a hipervizor koristi značajke hardverske arhitekture kako bi zaštitio fizičku memoriju i omogućio izolaciju između virtualnih strojeva. Operacijski sustav koji je pokrenut na vrhu hipervizora izvršava niti i obrađuje prekide na virtualnim procesorima, a hipervizor raspoređuje virtualne procesore na fizičkim procesorima. Hipervizor se i dalje izvodi s istom razinom privilegija CPU-a kao i jezgra, ali budući da koristi specijalizirane CPU upute, može se odvojiti od jezgre dok istovremeno nadgleda nju i aplikacije (Yosifovich i dr., 2017).

6.1. Korisnički prostor

Postoje četiri glavne vrste procesa u korisničkom načinu rada (Yosifovich i dr., 2017):

- Korisnički procesi – mogu biti jedni od sljedećih kategorija: Windows 32-bitni ili 64-bitni (u ovu kategoriju uključene su aplikacije koje se izvode u Windows Runtimeu u sustavu Windows 8 i svim kasnijim verzijama), Windows 3.1 16-bitni, MS-DOS 16-bitni, POSIX 32-bitni ili 64-bitni. Treba napomenuti da se 16-bitne aplikacije mogu pokretati samo na 32-bitnom sustavu Windows, a POSIX aplikacije više nisu podržane u novijim verzijama Windowsa.
- Uslužni procesi – predstavljaju procese koji hostiraju Windows usluge poput Print Spooler i Task Scheduler usluga. Usluge uglavnom imaju zahtjev da se pokreću neovisno o prijavama korisnika. Microsoft Exchange Server, Microsoft SQLServer i mnoge druge Windows poslužiteljske aplikacije također uključuju komponente koje se izvode kao usluge.
- Sistemski procesi – to su fiksni procesi kao što su Session Manager i proces prijave. Oni nisu Windows usluge, to jest ne pokreće ih upravitelj kontrole usluga (eng. Service Control Manager).

- Poslužiteljski procesi podsustava okoline – njihov zadatak je izložiti aplikacijskim programima neki podskup temeljnih usluga izvršnog sustava Windows. Svaki podsustav može dati pristup raznim podskupovima izvornih usluga Windowsa što znači da aplikacija izgrađena na jednom podsustavu može postići zadatke koje aplikacija na drugom podsustavu ne može. Dakle, ovi procesi implementiraju dio podrške ili osobnost za okolinu operacijskog sustava predstavljenu korisniku i programeru. Windows NT je prvobitno isporučen s tri podsustava okoline: Windows, POSIX i OS/2, međutim, podsustav OS/2 zadnji je put isporučen sa sustavom Windows 2000, a POSIX sa sustavom Windows XP. Ultimate i Enterprise izdanja sustava Windows 7, kao i sve poslužiteljske verzije Windowsa 2008 R2, uključuju podršku za poboljšani POSIX podsustav nazvan Podsustav (eng. Subsystem) za aplikacije temeljene na UNIX-u (SUA). SUA je ukinut i više se ne nudi kao opcionalni dio sustava Windows.

U operacijskom sustavu Windows, korisničke aplikacije ne pozivaju izravno matične Windows usluge, već one prolaze kroz jedan ili više podsustava knjižnice dinamičkih veza (DLL). DLL podsustavi imaju za zadatak prevesti dokumentirane funkcije u odgovarajuće interne ili nedokumentirane pozive matičnih usluga sustava koji su uglavnom implementirani u Ntdll.dll. Taj prijevod može, a i ne mora uključivati slanje poruke procesu podsustava okoline koji služi korisničkom procesu.

6.2. Prostor jezgre

Jezgreni način rada uključuje sljedeće komponente (Yosifovich i dr., 2017):

- Izvršne komponente (eng. executive) – sadrže osnovne usluge operacijskog sustava koje koriste svi podsustavi okoline. Neke od usluga su: umrežavanje, upravljanje memorijom, sigurnost, upravljanje procesima i nitima, upravljanje ulazno/izlaznim uređajima i slično.
- Jezgra Windowsa – sastoji se od funkcija operacijskog sustava niske razine poput višeprocesorske sinkronizacije, raspoređivanja niti, rukovanja prekidima i iznimkama te prebacivanja iz korisničkog načina rada u jezgreni način pomoću sučelja sistemskog poziva. Osim toga, omogućuje niz rutina i osnovne objekte koje ostale izvršne komponente koriste za implementaciju konstrukcija više razine.

- Upravljački programi – uključuju hardverske upravljačke programe koji pretvaraju pozive korisničkih ulazno/izlaznih funkcija u određene zahtjeve hardverskih ulazno/izlaznih uređaja. Osim njih, uključuju i upravljačke programe koji nisu dio hardvera poput datotečnog sustava i mrežnog upravljačkog programa.
- Sloj apstrakcije hardvera (HAL) – predstavlja sloj koda koji izolira jezgru, upravljačke programe i ostale izvršne komponente Windowsa od hardverskih razlika specifičnih za platformu kao što je razlika između matičnih ploča.
- Prozorski i grafički sustav – ovaj sustav implementira funkcije grafičkog sučelja (GUI), poznatije kao Windows USER i GDI funkcije kao što su crtanje, rad s prozorima i kontrola korisničkog sučelja.
- Sloj hipervizora – sastoji se od samo jedne komponente, a to je hipervizor. Ovaj sloj nema upravljačkih programa ili drugih modula. Sam hipervizor je sastavljen od više unutarnjih slojeva i usluga kao što su planiranje virtualnog procesora, vlastiti upravitelj memorije, sinkronizacijske rutine, upravljanje prekidima, upravljanje particijama poput virtualnih strojeva i drugo.

7. Rasprava

Ovaj završni rad ukratko opisuje glavne uloge i koncepte koji su dio svakog operacijskog sustava i bez kojih njegovo funkcioniranje ne bi bilo moguće. Svaki operacijski sustav se temelji na određenoj arhitekturi sastavljenoj od ključnih komponenti koje međusobno komuniciraju kako bi mogle izvršavati zadatke koje zatraže korisnici. Prema opisanim pojmovima i činjenicama vezanim uz operacijske sustave, mogu se iznijeti neke prednosti, a onda i nedostaci samih operacijskih sustava.

Operacijski sustav štedi vrijeme smanjenjem složenosti i pomaže poboljšati učinkovitost rada korisnika i stroja te se može lako ažurirati. Omogućuje lakše dijeljenje podataka s velikim brojem korisnika, a korisnici mogu lako instalirati i pokrenuti bilo koju aplikaciju. Pošto operacijski sustav djeluje kao sučelje između softvera i hardvera, korisnici mogu pristupiti hardveru bez pisanja velikih programa, a višezadaćnost postaje lakša. Također, postoje neki operacijski sustavi koji su otvorenog koda, odnosno mogu se besplatno pokrenuti na računalu. Ovo su samo neke od brojnih prednosti operacijskih sustava, a pošto sve ima i svoje loše strane, tako i operacijski sustavi imaju neke nedostatke. Za neke korisnike operacijski sustavi mogu biti dosta skupi. Kako je već navedeno, postoje sustavi otvorenog koda, no njih je malo teže pokrenuti od drugih. Još jedan nedostatak su virusi koji usporavaju i zaustavljaju rad operacijskog sustava. Neki operacijski sustavi su dosta složeni pa korisnici mogu imati poteškoća ako se pojavi problem u sustavu, a ako zbog bilo kojeg razloga operacijski sustav prestane funkcionirati, ruši se cijeli računalni sustav.

Kako bi se nedostaci sveli na minimum potrebno je izgraditi bolje operacijske sustave koji neće biti puni grešaka, nepouzdani i nesigurni. Teško je predvidjeti kamo ide računarstvo jer je to znanost koja jako brzo napreduje. Istraživači konstantno smišljaju nove ideje od kojih će neke imati veliki utjecaj na korisnike i industriju i biti temelj budućih proizvoda. U budućnosti se očekuje veliki pomak u odnosu na aktualne operacijske sustave. Jedno od glavnih obilježja će vjerojatno biti neometanje korisnika u smislu ažuriranja sustava, a neki sustavi su već riješili taj problem. Trebalo bi se također stvoriti sigurno okruženje za pokretanje aplikacija kako bi se zaštitio cijeli sustav. Osim toga, važna je i stalna povezanost modernog operacijskog sustava s internetom kao i održivost sustava u kojem korisnik neće brinuti je li baterija pri kraju ili ne (Čizmić, 2019). Što se računalnog sustava tiče, u arhitekturalnom pogledu razvoj će najvjerojatnije ići prema poboljšanju performansi, sigurnosti, energije te smanjenju troškova. Važno je da računalna arhitektura bude dizajnirana s odgovarajućim sučeljima i mehanizmima

jer je za postizanje navedenih poboljšanja potrebna opsežna softverska i hardverska suradnja (Hennessy, Patterson, 2019).

Pošto se u ovom radu pažnja posvećuje operacijskom sustavu Windows, onda je korisno napomenuti kako je Microsoft predstavio novi operacijski sustav Windows 11 koji će zamijeniti sadašnji Windows 10. Vidljive promjene su se pojavile u dizajnu koji je postao jednostavniji, ikone aplikacije na programskoj traci su stavljene u središte što podsjeća na Mac OS, a također se koriste i zaobljeni kutovi prozora te su osvježene ikone i pozadine. Najveća promjena je što će Microsoft Store podržavati Android aplikacije koje se mogu preuzeti putem Amazonove trgovine. Poboljšane su i značajke višezadaćnosti kao i značajke vezane uz igre. Microsoft tvrdi da će Windows 11 biti brži od Windowsa 10, što znači brže pokretanje sustava i poboljšanu brzinu za web preglednike. Osim toga, bit će brža i ažuriranja koja mogu biti neprimjetna i događati se automatski u pozadini, a Microsoft sugerira da Windows 11 nudi i veću učinkovitost trajanja baterije za prijenosna računala. Glavni nedostatak ovog sustava je što ima visoke sistemske zahtjeve što znači da sustavi sa starijim procesorima možda neće biti kompatibilni (Jones, 2021).

8. Zaključak

Operacijski sustav je softver koji upravlja hardverom računala, a osim toga, pruža osnovu za aplikacijske programe i djeluje kao posrednik između korisnika računala i računalnog hardvera. Operacijski sustavi su svugdje prisutni, od dizala, automobila i kućanskih aparata do pametnih telefona te osobnih i poslovnih računala. Bez operacijskih sustava nije moguće pokrenuti druge programe pa korisnik ne može raditi u računalu i zato operacijski sustav ima jako važnu ulogu u računalnom sustavu. Važno je razumjeti organizaciju i arhitekturu računalnog hardvera kako bi se definirala uloga operacijskog sustava u modernom računalnom okruženju. To uključuje procesor, memoriju i ulazno/izlazne uređaje kao i pohranu, a glavna odgovornost operacijskog sustava je dodjela tih resursa programima. Mora se kreirati dio po dio operacijskog sustava budući da je to dosta velik i složen sustav, a svaki od dijelova bi trebao biti dobro definirani dio sustava s pažljivo definiranim funkcijama.

Korištenjem usporednih arhitekturnih rješenja ostvaruje se povećanje moći računalnih sustava. Prema tome, važno je poznavanje rada sklopovlja i mogućnosti koje se iskorištavaju kroz sučelje operacijskog sustava kako bi se kvalitetna programska podrška mogla izgraditi, a koja bi efikasno iskorištavala raspoložive resurse. U ovom radu su navedene neke glavne arhitekture suvremenih operacijskih sustava koje daju ideju o određenim izgledima koji su isprobani u praksi, a ujedno se dobiva predodžba o rasponu mogućnosti. Kako bi bila jasnija sama arhitektura, jedno poglavlje je posvećeno arhitekturi operacijskog sustava Microsoft Windows u kojem su ukratko objašnjene ključne komponente, način međusobne interakcije i kontekst u kojem rade. Budući da se tehnologija brzo razvija, za očekivati je kako se predviđaju promjene u odnosu na sadašnje operacijske sustave. Današnji operacijski sustavi su usmjereni prema korisniku, a svaka nova verzija donosi jednostavnije korištenje njihovih mogućnosti.

Literatura

1. AfterAcademy (2019). What are the types of an Operating System?. Dostupno na: https://afteracademy.com/blog/what-are-the-types-of-an-operating-system [pristupljeno 16. lipnja 2021.].
2. Budin, L., Golub, M., Jakobović, D., Jelenković, L. (2010). *Operacijski sustavi.* [Online] Dostupno na: https://vdocuments.site/operacijski-sustavi-586e2e1611259.html [pristupljeno 20. lipnja 2021.].
3. Čizmić, M. (2019). Microsoftov potpredsjednik opisuje "moderni operativni sustav" budućnosti koji jako podsjeća na Googleov Chrome OS. Dostupno na: https://zimo.dnevnik.hr/clanak/microsoftov-potpredsjednik-opisuje-moderni-operativni-sustav-buducnosti-koji-jako-podsjeca-na-googleov-chrome-os---562908.html [pristupljeno 30. lipnja 2021.].
4. Dmitrović, A. (2018). Adresiranje sektora na diskovima. Dostupno na: https://sysportal.carnet.hr/node/1799 [pristupljeno 14. srpnja 2021.].
5. Hennessy, L. J., Patterson, A. D. (2019). A New Golden Age for Computer Architecture. Dostupno na: https://cacm.acm.org/magazines/2019/2/234352-a-new-golden-age-for-computer-architecture/fulltext?mobile=false [pristupljeno 16. srpnja 2021.].
6. Jelenković, L. (2020). *Operacijski sustavi.* [Online] Dostupno na: http://www.zemris.fer.hr/~leonardo/os/fer/_OS-skripta.pdf [pristupljeno 10. lipnja 2021.].
7. Jones, R. (2021). Everything you need to know about Windows 11. Dostupno na: https://www.trustedreviews.com/news/windows-11-release-date-price-leaks-and-features-4145339 [pristupljeno 17. srpnja 2021.].
8. Mesarić, J., Zekić-Sušac, M., Dukić, B. (2015). Sistemski softver, predavanja na predmetu Informatika. Ekonomski fakultet u Osijeku, ak.god. 2015/16. Dostupno na: http://www.efos.unios.hr/informatika/wp-content/uploads/sites/202/2013/04/Sistemski_softver_2015.pdf [pristupljeno 13. lipnja 2021.].

9. PadaKuu (2021). Architectures of Operating System. Dostupno na: https://padakuu.com/article/37-architectures-of-operating-system [pristupljeno 21. lipnja 2021.].

10. Russinovich, M., Solomon, D. A., Ionescu, A. (2012). *Windows Internals.* [Online] Dostupno na: https://repo.zenk-security.com/Linux%20et%20systemes%20d.exploitations/Windows%20Internals%20Part%201_6th%20Edition.pdf [pristupljeno 27. lipnja 2021.].

11. Silberschatz, A., Galvin, P. B., Gagne, G. (2018). *Operating System Concepts.* [Online] Dostupno na: http://www.nastooh.com/teaching/Silberschatz_Operating_System_Concepts_10e_2018.pdf0Part%201_6th%20Edition.pdf [pristupljeno 26. lipnja 2021.].

12. Studytonight (2021). Types of Operating Systems. Dostupno na: https://www.studytonight.com/operating-system/types-of-os [pristupljeno 15. lipnja 2021.].

13. Tanenbaum, A. S., Bos, H. (2015). *Modern Operating Systems.* [Online] Dostupno na: http://index-of.es/Varios-2/Modern%20Operating%20Systems%204th%20Edition.pdf [pristupljeno 22. lipnja 2021.].

14. Tutorialspoint (2021). Types of Operating Systems. Dostupno na: https://www.tutorialspoint.com/operating_system/os_types.htm [pristupljeno 15. lipnja 2021.].

15. Vukelić, B. (n.d.). Operacijski sustavi, interni materijali na predmetu Operacijski sustavi. Veleučilište u Rijeci. Dostupno na: https://www.veleri.hr/files/datotekep/nastavni_materijali/k_informatika_1/Predavanja_0102.pdf [pristupljeno 19. lipnja 2021.].

16. Yosifovich, P., Ionescu, A., Russinovich, M. E., Solomon, D. E., (2017). *Windows Internals.* [Online] Dostupno na: http://index-of.es/Varios-2/Windows%20System%20Internals%20Part%201.pdf [pristupljeno 29. lipnja 2021.].

Popis slika