

Pregled operacijskih sustava



Operacijski sustav

- Program koji kontrolira izvođenje aplikacijskih programa
- Sučelje između aplikacija i hardvera

Glavni ciljevi OS-a:

- Pogodnost
- Učinkovitost
- Sposobnost razvoja

Usluge operacijskog sustava

- Razvoj programa
- Izvršenje programa
- Pristup U/I uređajima
- Kontrolirani pristup datotekama
- Pristup sustavu
- Otkrivanje grešaka i odgovor
- Praćenje aktivnosti

Ključna sučelja

- Arhitektura skupa instrukcija (ISA)
- Binarno sučelje aplikacije (ABI)
- Sučelje za programiranje aplikacije (API)

Operativni sustav - upravitelj resursima

- OS je odgovoran za kontrolu korištenja resursa računala, kao što su I/O, glavna i sekundarna memorija i vrijeme izvršavanja procesora
- Funkcionira na isti način kao i obični računalni softver
- Program ili skup programa koje izvršava processor
- Često kontrolu prepušta procesoru i čeka na ponovno preuzimanje kontrole

Evolucija operativnih sustava

- Operacijski sustavi se vremenom razvijaju iz nekoliko razloga:

Nadogradnja hardwarea

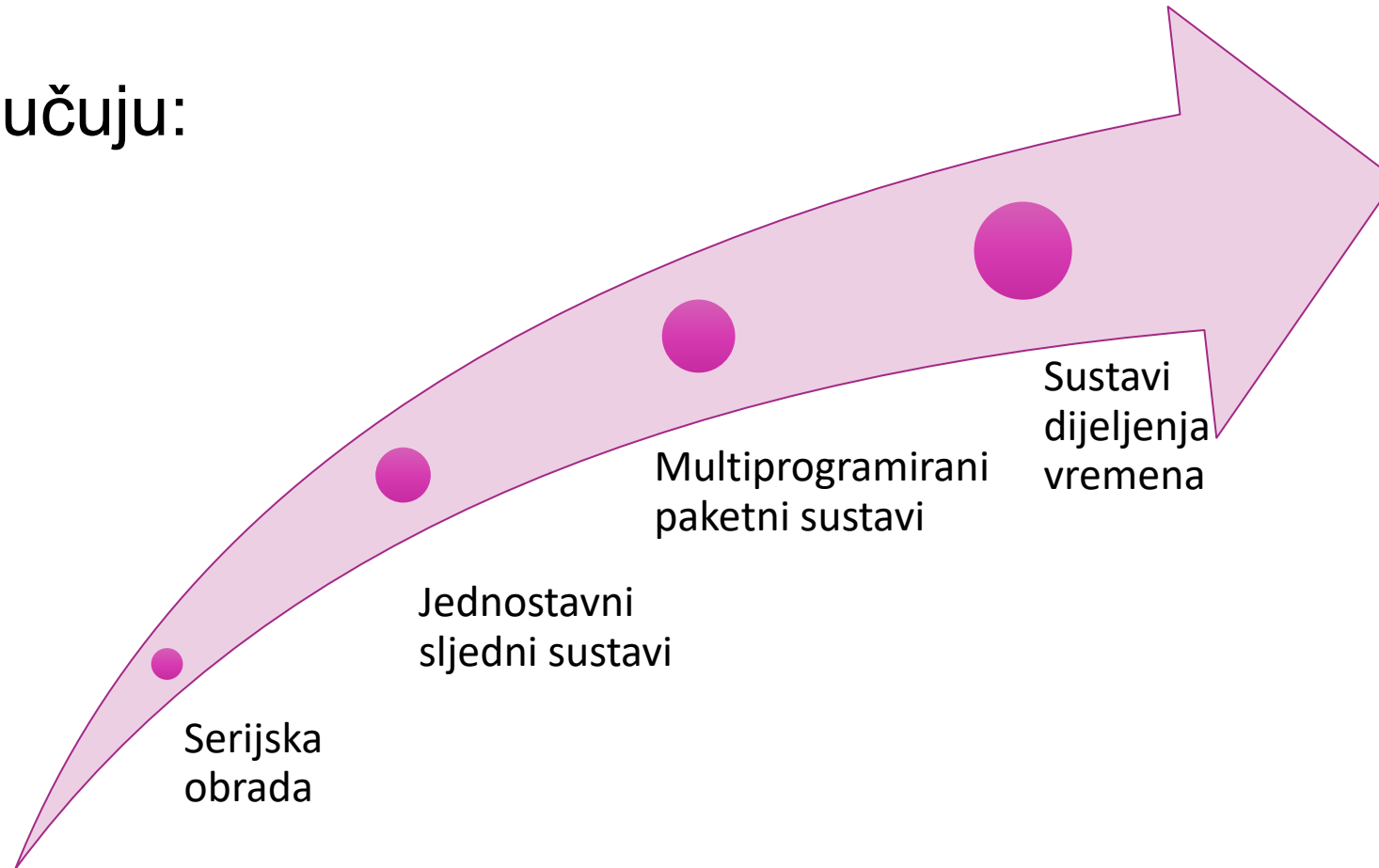
Novi tipovi hardwarea

Nove usluge

Popravci

Evolucija operativnih sustava

- Faze uključuju:



Serijska obrada

Najranija računala:

- Nema operacijskog sustava
 - Programeri su izravno komunicirali s računalnim hardverom
- Računala su radila s konzole sa svjetlima zaslona, prekidačima, nekim oblikom uređaja za unos i pisačem
- Korisnici imaju pristup računalu u "seriji"

Problemi:

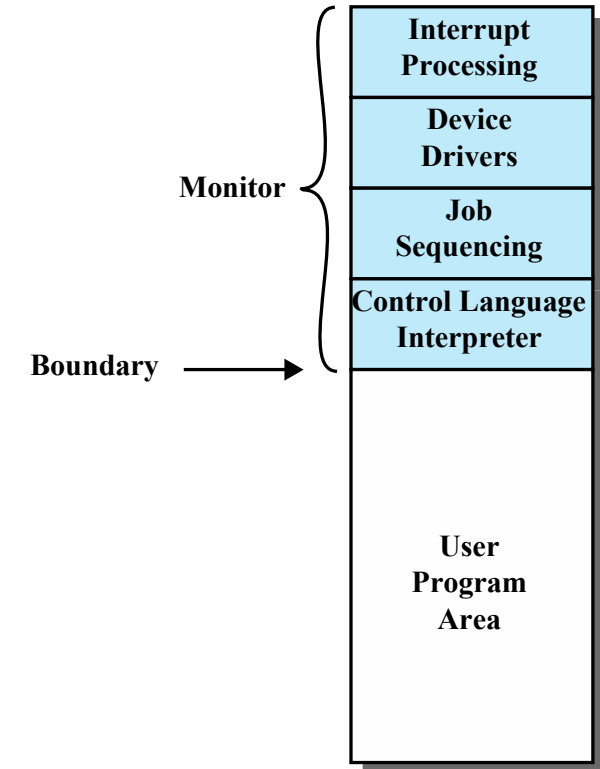
- Zakazivanje:
 - Većina instalacija koristila je tiskani list za prijavu za rezerviranje vremena za računalno
 - Dodjela vremena mogla bi trajati kratko ili dugo, što bi rezultiralo gubitkom vremena na računalu
- Vrijeme postavljanja
 - Znatna količina vremena potrošena je na postavljanje programa za rad

Jednostavni sljedni sustavi

- Rana računala bila su vrlo skupa
 - Važno za maksimalno korištenje procesora
- Monitor
 - Korisnik više nema izravan pristup procesoru
 - Posao se predaje računalnom operateru koji ih spaja i stavlja na uređaj za unos
 - Program se grana natrag na monitor kada završi

Točka gledišta monitora

- Monitor kontrolira slijed događaja
- Stalni monitor je softver koji je uvijek u memoriji
- Monitor očitava posao i daje kontrolu
- Zadatak vraća kontrolu nadzoru



Točka gledišta procesora

- Procesor izvršava instrukcije iz memorije koja sadrži monitor
- Izvršava upute u korisničkom programu sve dok ne naiđe na uvjet završetka ili pogreške
- "Kontrola je proslijeđena poslu" znači da procesor dohvaća i izvršava upute u korisničkom program
- "Kontrola je vraćena monitoru" znači da procesor dohvaća i izvršava upute iz programa monitora

Jezik kontrole posla (Job Control Language - JCL)

Posebna vrsta programskog jezika
koji se koristi za davanje uputa
monitoru

Koji prevodilac koristiti

Koje podatke koristiti

Poželjne hardverske značajke

Zaštita memorije

- Dok se korisnički program izvršava, ne smije mijenjati memorijsko područje koje sadrži monitor

Tajmer

- Sprječava da posao monopolizira sustav

Privilegirane instrukcije

- Mogu biti izvršene samo od monitora

Prekidi

- OS-u daje veću fleksibilnost u kontroli korisničkih programa

Načini rada

Korisnički način rada

- Korisnički program se izvršava u korisničkom načinu rada
- Određena područja memorije zaštićena su od pristupa korisnika
- Određene instrukcije se neće izvršiti

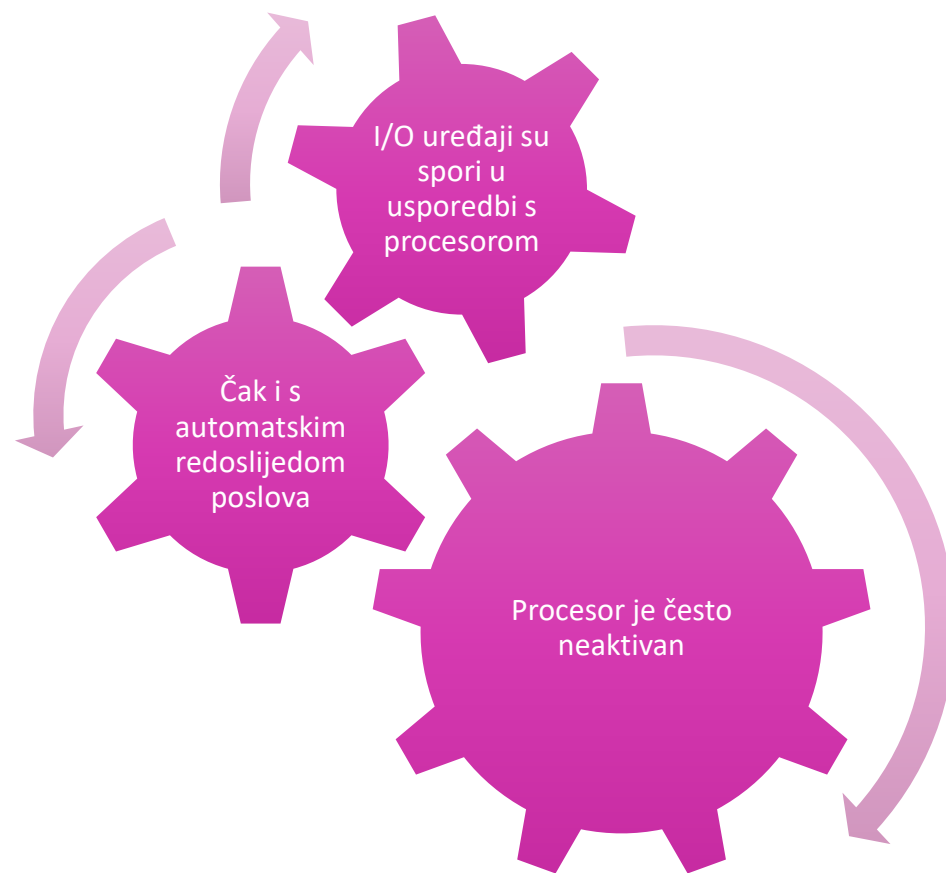
Jezgrin način rada

- Monitor se izvršava u jezgrinom modu
- Privilegirane instrukcije mogu se izvršiti
- Može se pristupiti zaštićenim područjima memorije

Nedostaci jednostavnog sljednog sustava

- Vrijeme procesora izmjenjuje se između izvršavanja korisničkih programa i izvršavanja monitora
- Dio glavne memorije sada je prepušten monitoru
- Monitor troši određeno vrijeme procesora
- Unatoč dodatnim gubitcima, jednostavan sljedni sustav poboljšava korištenje računala

Multiprogramirani paketni sustavi



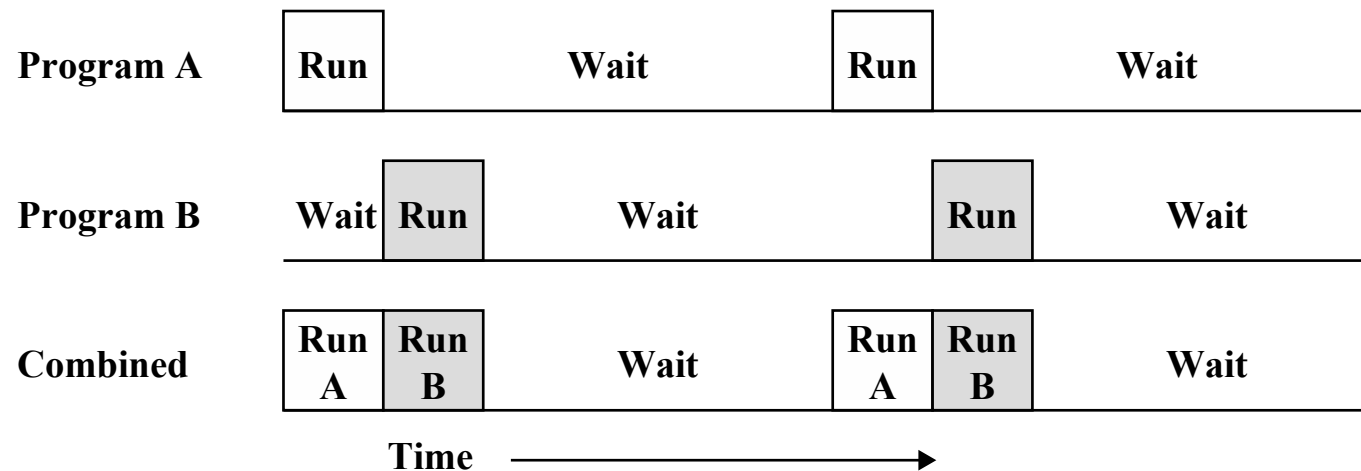
Uniprogramiranje

- Procesor troši određeno vrijeme na izvršavanje, sve dok ne dođe do I/O instrukcije; tada mora pričekati dok se ta I/O instrukcija ne završi prije nego što nastavi



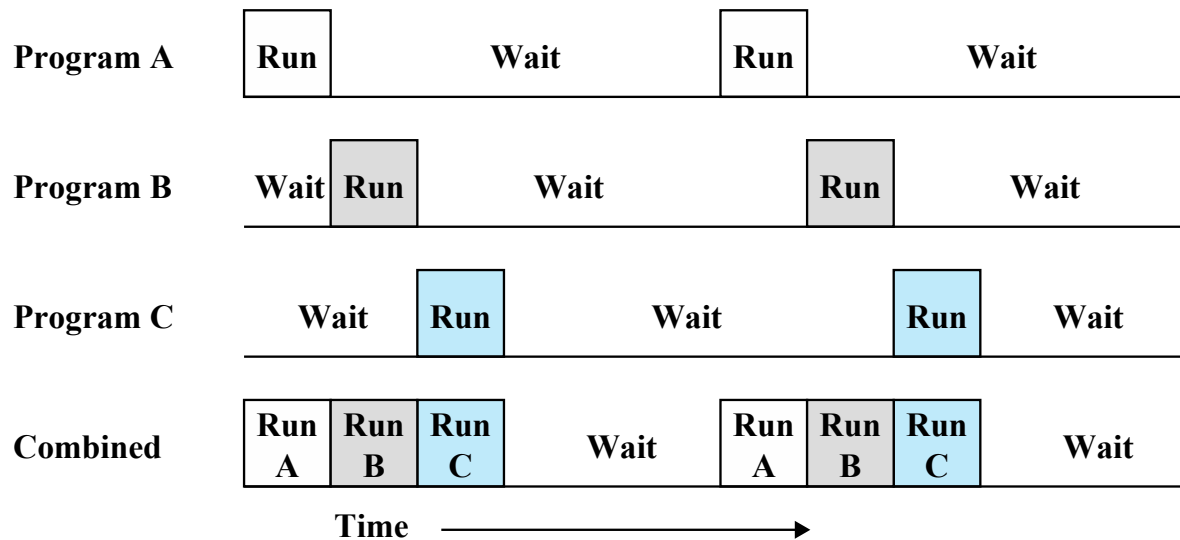
Multiprogramiranje

- Mora postojati dovoljno memorije za OS (rezidentni monitor) i jedan korisnički program
- Kada jedan posao treba čekati na I/O, procesor se može prebaciti na drugi posao, koji vjerojatno ne čeka na I/O



Multiprogramiranje - višezadačnost

- Također poznat kao višezadačnost
- Memorija se proširuje za držanje tri, četiri ili više programa i prebacivanje između svih njih



Sustavi dijeljenja vremena

- Može se koristiti za rukovanje više interaktivnih poslova
- Vrijeme procesora dijeli se među više korisnika
- Više korisnika istovremeno pristupa sustavu putem terminala, pri čemu OS isprepliće izvršavanje svakog korisničkog programa u kratkom nizu ili kvantnom izračunavanju

Kompatibilni sustav dijeljenja vremena (CTSS)

- Jedan od prvih operativnih sustava za dijeljenje vremena
- Razvijen na MIT-u od strane grupe poznate kao Project MAC
- Sustav je prvi put razvijen za IBM 709 1961. godine

Kompatibilni sustav dijeljenja vremena (CTSS)

- Rad na računalu s 32.000 36-bitnih riječi glavne memorije, a stalni monitor troši 5.000 od toga
- Koristio je tehniku poznatu kao rezanje vremena
 - Sistemski sat generira prekide brzinom od otprilike jedan svake 0,2 sekunde
 - Na svakom prekidu takta OS je ponovno preuzeo kontrolu i mogao je dodijeliti procesor drugom korisniku
 - Tako bi u redovitim vremenskim intervalima trenutni korisnik bio preuzet, a drugi bi učitavan
 - Kako bi se sačuvao status starog korisničkog programa za kasniji nastavak, stari korisnički programi i podaci zapisani su na disk prije nego što su novi korisnički programi i podaci pročitani
 - Stari korisnički programski kod i podaci vraćeni su u glavnu memoriju kada je taj program sljedeći put bio uključen

Glavna postignuća

- Operativni sustavi su među najsloženijim
- Glavni napredak u razvoju uključuje:
 - Procesi
 - Upravljanje memorijom
 - Zaštita i sigurnost informacija
 - Planiranje i upravljanje resursima
 - Struktura sustava

Proces

- Temeljna struktura operacijskih sustava

Proces se može definirati kao:

Program u izvođenju

Instanca pokrenutog programa

Entitet koji se može dodijeliti procesoru i izvršiti na njemu

Jedinica aktivnosti koju karakterizira jedna sekvencijalna nit izvršenja, trenutno stanje i pridruženi skup resursa sustava

Uzroci pogrešaka

- Nepravilna sinkronizacija
 - Često je slučaj da se rutina mora obustaviti čekajući događaj negdje drugdje u sustavu
 - Nepravilan dizajn signalnog mehanizma može rezultirati gubitkom ili dupliciranjem
- Neuspješno međusobno isključivanje
 - Više od jednog korisnika ili programa pokušava koristiti zajednički resurs u isto vrijeme
 - Mora postojati neka vrsta mehanizma međusobnog isključivanja koji dopušta samo jednoj rutini u isto vrijeme da izvrši ažuriranje datoteke
- Neodređeni programski rad
 - Kada programi dijele memoriju, a njihovo izvršavanje se isprepliće od strane procesora, oni mogu ometati jedni druge prepisivanjem zajedničkih memorijskih područja na nepredvidive načine
 - Redoslijed rasporeda programa može utjecati na ishod svakog pojedinog programa
- Zastoji

Moguće je da dva ili više programa čekaju jedan na drugog

Komponente procesa

- Proces se sastoji od tri komponente:
 - Izvršni program
 - Pridruženi podaci potrebni programu (varijable, radni prostor, međuspremници, itd.)
 - Kontekst izvršenja (ili “stanje procesa”) procesa
- Kontekst izvršenja je bitan:
 - To su interni podaci pomoću kojih OS može nadzirati i kontrolirati process
 - Uključuje sadržaje različitih registara procesa
 - Uključuje informacije kao što su prioritet procesa i čeka li proces na završetak određenog I/O događaja

Upravljanje memorijom

- OS ima pet glavnih odgovornosti za upravljanje pohranom:

Izolacija
procesa

Automatska
dodjela i
upravljanje

Podrška
modularnom
programiranju

Zaštita i
kontrola
pristupa

Dugotrajno
skladištenje

Virtualna memorija

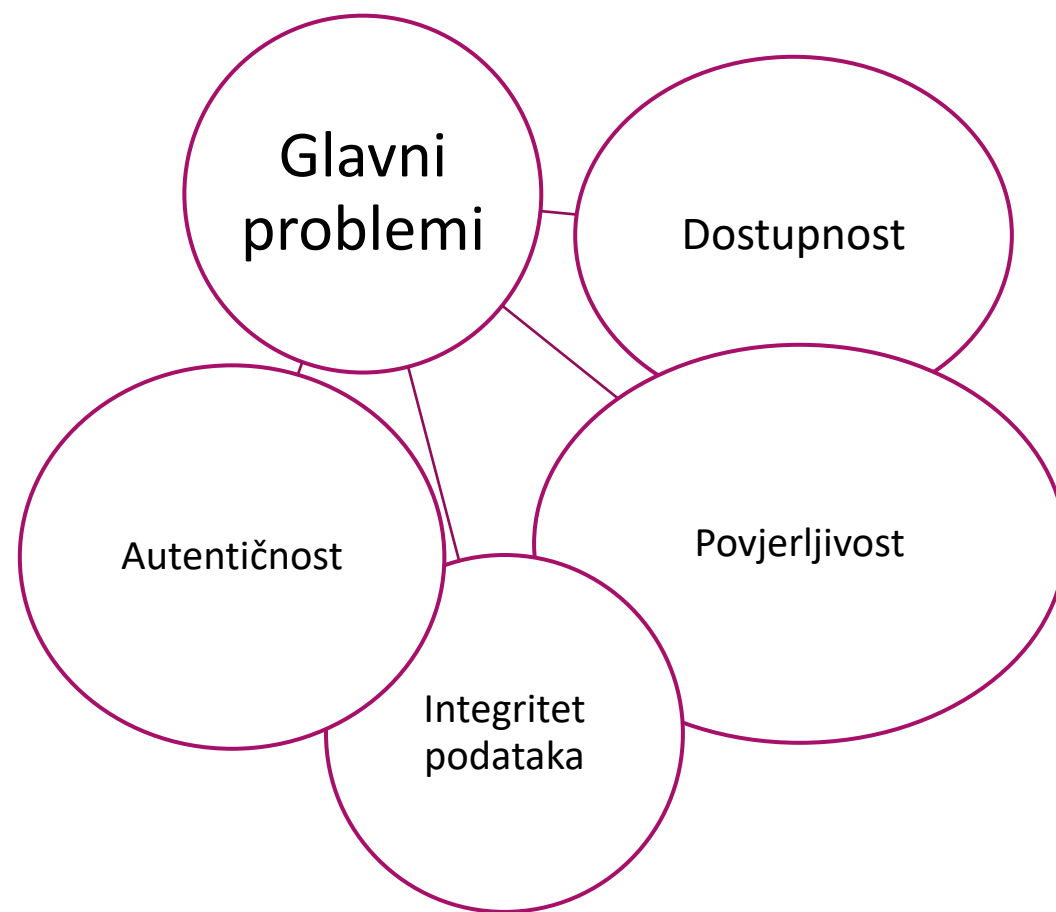
- Sustav koji omogućuje procesima da adresiraju memoriju s logičke točke gledišta, bez obzira na količinu glavne memorije koja je fizički dostupna
- Zamišljeno da zadovolji zahtjev da više korisničkih poslova istovremeno boravi u glavnoj memoriji

Straničenje

- Omogućuje da se procesi sastoje od niza blokova fiksne veličine, koji se nazivaju stranicama
- Program upućuje na riječ pomoću virtualne adrese, koja se sastoji od broja stranice i pomaka unutar stranice
- Svaka stranica procesa može se nalaziti bilo gdje u glavnoj memoriji
- Sustav straničnog poziva omogućuje dinamičko preslikavanje između virtualne adrese koja se koristi u procesu i stvarne adrese (ili fizičke adrese) u glavnoj memoriji

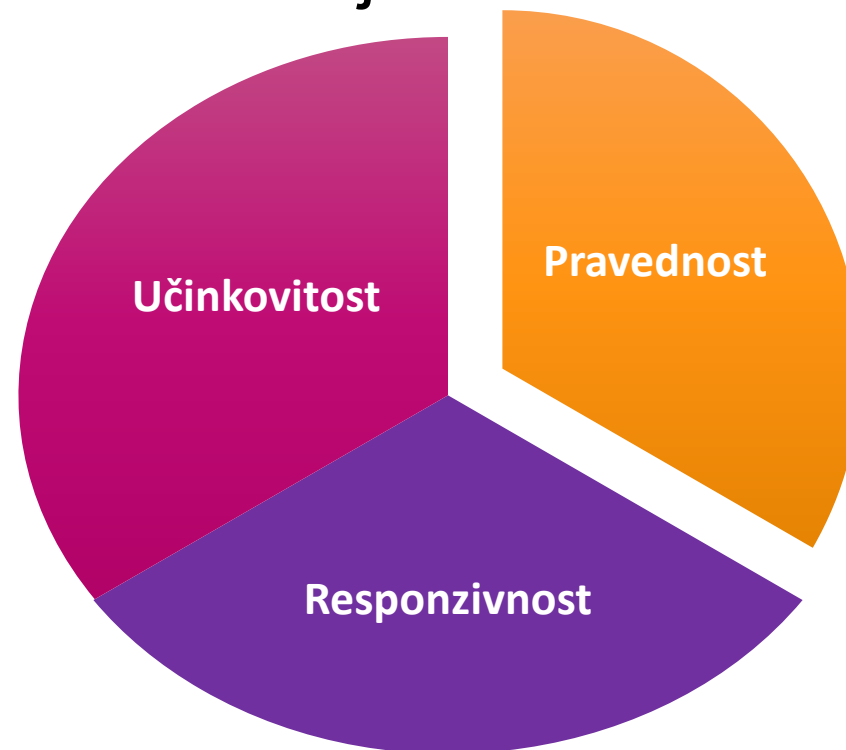
Zaštita i sigurnost informacija

- Priroda prijetnje koja se tiče organizacije uvelike će varirati ovisno o okolnostima
- Problem uključuje kontrolu pristupa računalnim sustavima i informacijama pohranjenim u njima



Planiranje i upravljanje resursima

- Ključna odgovornost OS-a je upravljanje resursima
- Politike dodjele resursa moraju uzeti u obzir:



Različiti arhitektonski pristupi

- Zahtjevi za operativnim sustavima zahtijevaju nove načine organiziranja OS-a

Isprobani su različiti pristupi i elementi dizajna:

- Mikrokernel arhitektura
- Višedretvenost
- Simetrična višeprocena obrada
- Distribuirani operativni sustavi
- Objektно orijentirani dizajn

Microkernel Architecture

- Dodjeljuje samo nekoliko bitnih funkcija kernelu:

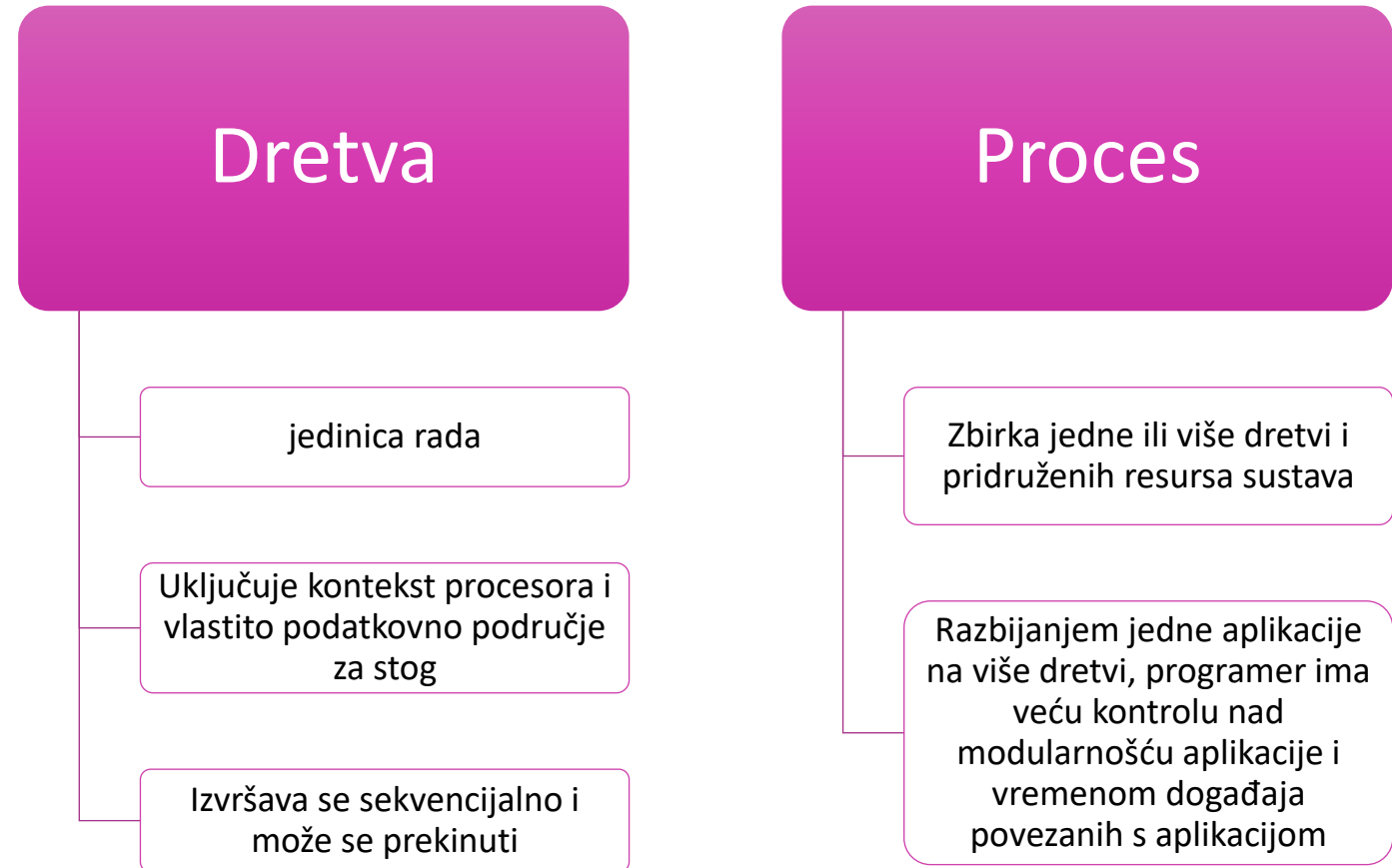


- Pristup:



Višedretvenost

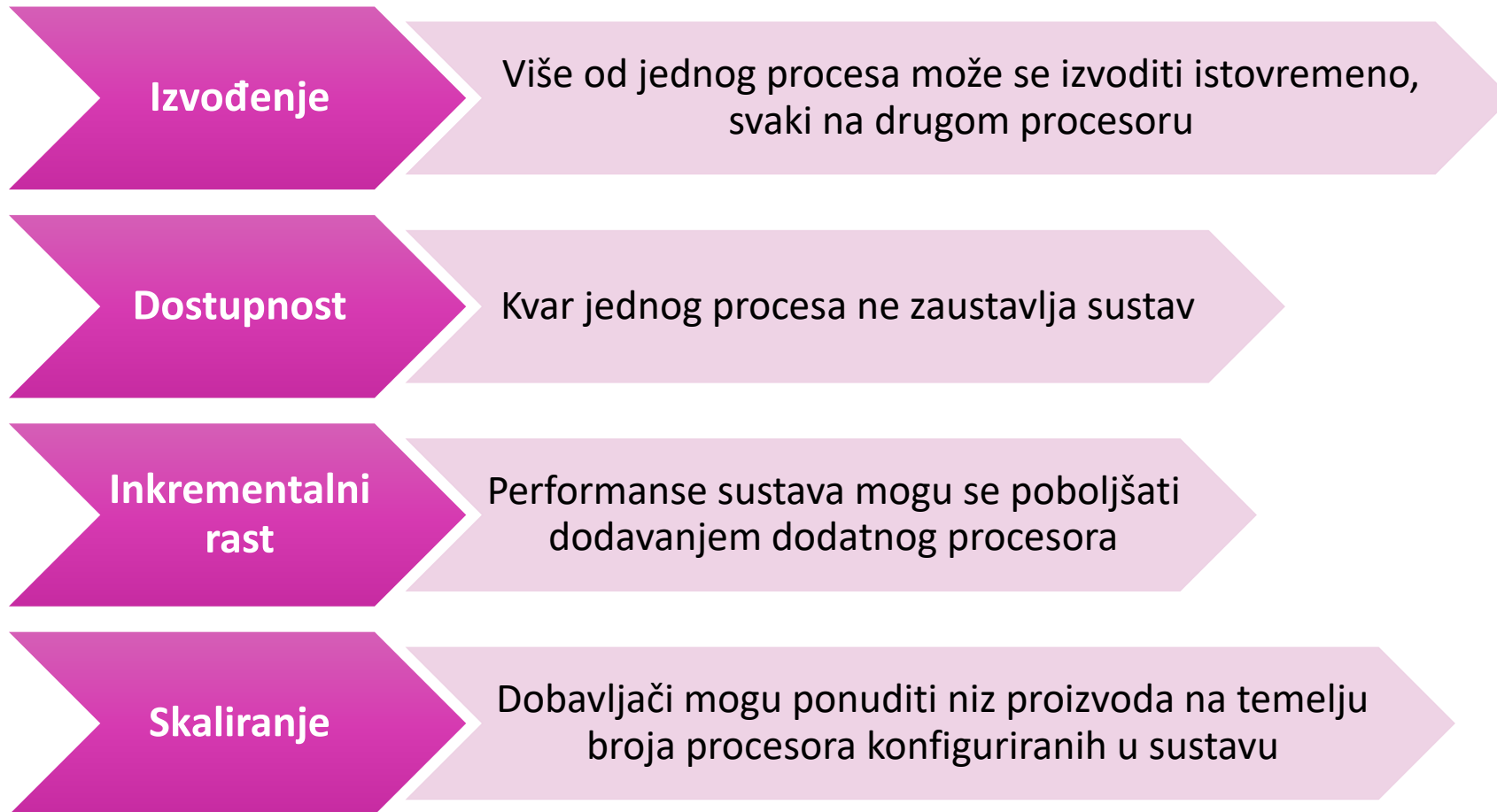
- Tehnika u kojoj se proces, izvršavajući aplikaciju, dijeli na niti koje se mogu izvoditi istodobno



Simetrična višeprocenska obrada (SMP)

- Pojam koji se odnosi na arhitekturu računalnog hardvera i također na ponašanje OS-a koje iskorištava tu arhitekturu
- OS SMP-a raspoređuje procese ili dretve u svim procesorima
- OS mora osigurati alate i funkcije za iskorištavanje paralelizma u SMP sustavu
- O višedretvenosti i SMP-u se često raspravlja zajedno, ali to su dva neovisna mehanizma
- Postojanje više procesora transparentno je za korisnika

Prednosti SMP-a



Dizajn OS-a

Distribuirani operativni sustav

- Pruža iluziju jednog glavnog memorijskog prostora i jednog sekundarnog memorijskog prostora plus druge objekte ujedinjenog pristupa, kao što je distribuirani datotečni sustav
- Najnoviji sustav za distribuirane operacijske sustave zaostaje za jednoprocorskim i SMP operativnim sustavima

Objektno orijentirani dizajn

- Daje disciplinu procesu dodavanja modularnih proširenja maloj kernel
- Omogućuje programerima da prilagode operativni sustav bez narušavanja integriteta sustava
- Također olakšava razvoj distribuiranih alata i cjelovitih distribuiranih operativnih sustava

Otpornost na greške

- Odnosi se na sposobnost sustava ili komponente da nastavi normalan rad unatoč prisutnosti hardverskih ili softverskih grešaka
- Obično uključuje određeni stupanj redundancije
- Namijenjen je povećanju pouzdanosti sustava
 - Obično dolazi s troškom u financijskom smislu ili smanjenju performansi
- Opseg usvajanja mjera tolerancije grešaka mora se odrediti prema tome koliko je resurs kritičan

Temeljni koncepti

- Osnovne mjere su:
 - Pouzdanost
 - $R(t)$
 - Definira se kao vjerojatnost njegovog ispravnog rada do vremena t s obzirom da je sustav ispravno radio u vrijeme $t=0$
 - Srednje vrijeme do kvara (MTTF)
 - Srednje vrijeme za popravak (MTTR) je prosječno vrijeme potrebno za popravak ili zamjenu neispravnog elementa
 - Dostupnost
 - Definirano kao dio vremena u kojem je sustav dostupan za usluživanje zahtjeva korisnika

Greške

- Definirani su u IEEE standardu kao pogrešno stanje hardvera ili softvera koje proizlazi iz:
 - Kvar komponente
 - Greška operatera
 - Fizičke smetnje iz okoline
 - Pogreška u dizajnu
 - Programska pogreška
 - Greška strukture podataka
- Standard također navodi da se kvar očituje kao:
 - Kvar u hardverskom uređaju ili komponenti
 - Netočna definicija koraka, procesa ili podataka u računalnom programu

Kategorije grešaka

- Trajna
 - Greška koja je, nakon što se pojavi, uvijek prisutna
 - Greška traje sve dok se neispravna komponenta ne zamijeni ili popravi
- Privremena
 - Greška koja nije prisutna cijelo vrijeme za sve radne uvjete
 - Može se klasificirati kao
 - Prolazna – kvar koji se javlja samo jednom
 - Povremena – kvar koji se javlja u više nepredvidivih trenutaka

Metode redundantnosti

Prostorna (fizička) redundancija

Uključuje korištenje više komponenti koje ili obavljaju istu funkciju istovremeno ili su konfigurirane tako da je jedna komponenta dostupna kao rezervna kopija u slučaju kvara druge komponente

Vremenska redundancija

Uključuje ponavljanje funkcije ili operacije kada se otkrije pogreška

Djelotvoran je kod privremenih kvarova, ali nije koristan za trajne kvarove

Informacijska redundancija

Pružuje toleranciju grešaka repliciranjem ili kodiranjem podataka na takav način da se pogreške bitova mogu otkriti i ispraviti

Mehanizmi operacijskog sustava

- Brojne tehnike mogu se ugraditi u OS za podršku tolerancije grešaka:
 - Izolacija procesa
 - Kontrole istodobnosti
 - Virtualni strojevi
 - Kontrolne točke i vraćanja

Razmatranja simetričnog višeprosesorskog OS-a

- Višeprosesorski OS mora pružiti svu funkcionalnost sustava za više programiranja plus dodatne značajke za smještaj više procesora
- Ključni problemi dizajna:

Istovremeni konkurentni procesi ili dretve

Rutine kernela moraju biti ponovno uključene kako bi se omogućilo da nekoliko procesora istovremeno izvršava isti kod kernela

Dodjela vremena

Bilo koji procesor može izvršiti planiranje, što komplicira zadatak provođenja politike planiranja

Sinhronizacija

S više aktivnih procesa koji imaju potencijalni pristup zajedničkim adresnim prostorima ili zajedničkim I/O resursima, mora se voditi računa o pružanju učinkovite sinkronizacije

Upravljanje memorijom

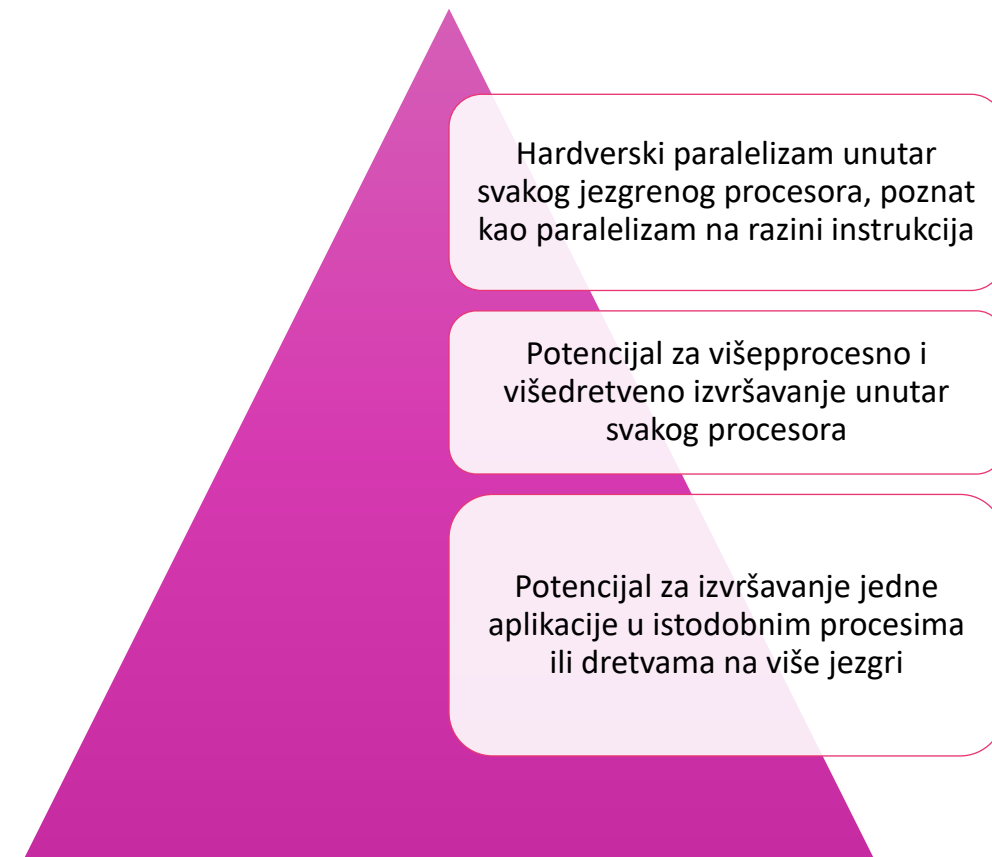
Ponovna upotreba fizičkih stranica

Pouzdanost i tolerancija na greške

OS bi trebao osigurati degradaciju u slučaju kvara procesora

Razmatranja višejezgrenog OS-a

- Izazov dizajna za višejezgreni sustav je učinkovito iskoristiti višejezgrenu procesorsku snagu i inteligentno učinkovito upravljati značajnim resursima na čipu
- Potencijal za paralelizam postoji na tri razine:



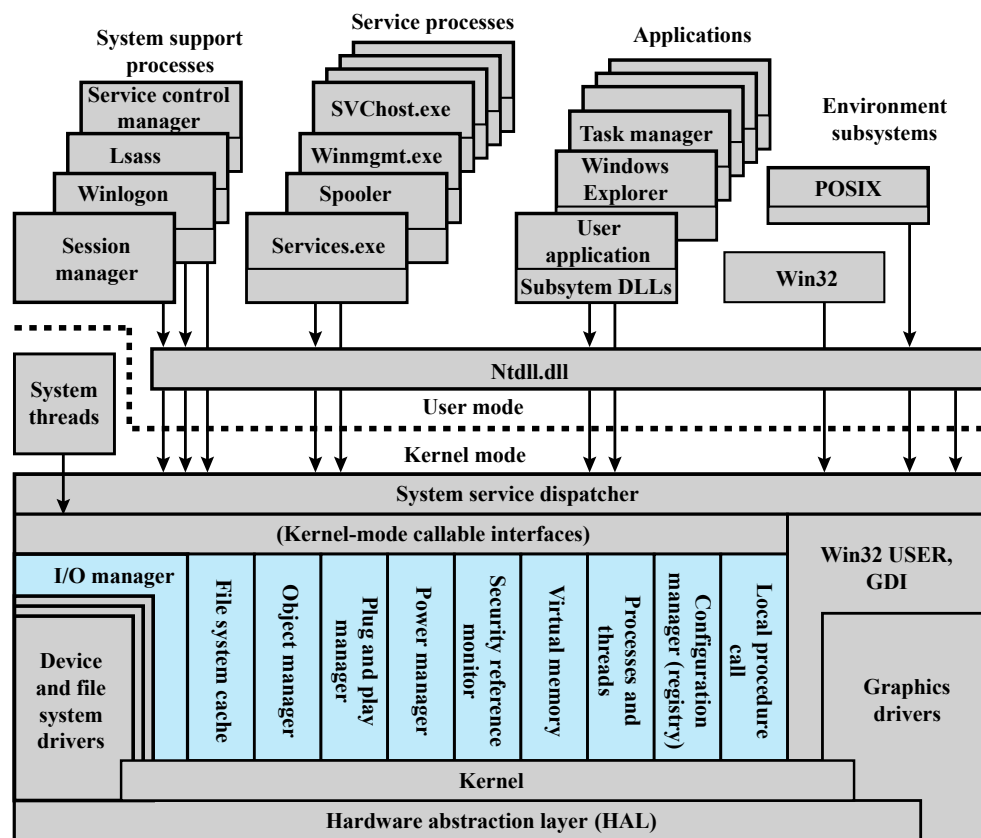
Grand Central Dispatch (GCD)

- Mogućnost višejezgrene podrške
 - Nakon što je programer identificirao nešto što se može odvojiti u zaseban zadatak, GCD to raspoređuje po jezgrama.
- U suštini, GCD je mehanizam skupa dretvii, u kojem OS preslikava zadatke na dretve koje predstavljaju dostupni stupanj istodobnosti
- Pruža proširenje programskih jezika kako bi se omogućilo pozivanje funkcija kao način određivanja zadataka
- Olakšava djeljenje jedinica rada uz održavanje eda i ovisnosti između podzadataka

Virtualna računala

- Omogućuje da se jedna ili više jezgri posvete određenom procesu, a zatim ostavi procesor na miru da posveti svoje napore tom procesu
- Višejezgreni OS bi tada mogao djelovati kao hipervizor koji donosi odluku na visokoj razini o dodjeli jezgri aplikacijama

Arhitektura Windows OS-a



Lsass = local security authentication server
POSIX = portable operating system interface
GDI = graphics device interface
DLL = dynamic link libraries

Colored area indicates Executive

Komponente jezge Windows OS-a

- Izvršni dio
 - Sadrži osnovne OS usluge, kao što su upravljanje memorijom, upravljanje procesima i dretvama, sigurnost, I/O i međuprocena komunikacija
- Jezgra
 - Kontrolira rad procesora. Kernel upravlja raspoređivanjem dretvi, prebacivanjem procesa, rukovanjem iznimkama i prekidima te sinkronizacijom više procesora
- Sloj hardverske apstrakcije (HAL)
 - Preslikava između generičkih hardverskih naredbi i odgovora i onih jedinstvenih za određenu platformu i izolira OS od hardverskih razlika specifičnih za platformu
- Upravljački program
 - Dinamičke biblioteke koje proširuju funkcionalnost izvršnog djela. To uključuje upravljačke programe hardverskih uređaja koji prevode pozive korisničkih I/O funkcija u specifične zahtjeve za I/O hardverskog uređaja i softverske komponente za implementaciju datotečnih sustava, mrežnih protokola i svih drugih proširenja sustava koja se trebaju izvoditi u načinu rada jezgre
- Prozorski i grafički sustav
 - Implementira GUI funkcije, kao što su rad s prozorima, kontrole korisničkog sučelja i crtanje

Izvršni dio Windows OS-a

I/O upravitelj

Pruža okvir kroz koji su I/O uređaji dostupni aplikacijama i odgovoran je za slanje odgovarajućim upravljačkim programima uređaja za daljnju obradu

Upravitelj predmemorije

Poboljšava performanse I/O-a temeljenog na datoteci tako što uzrokuje da se nedavno referencirani podaci datoteke nalaze u glavnoj memoriji radi brzog pristupa i odgađanjem pisanja na disk zadržavanjem ažuriranja u memoriji kratko vrijeme prije nego što ih pohrani

Upravitelj objekata

Stvara, upravlja i briše Windows Executive objekte koji se koriste za predstavljanje resursa kao što su procesi, dretve i sinkronizacijski objekti te provodi jedinstvena pravila za zadržavanje, imenovanje i postavljanje sigurnosti objekata

Plug-and-play upravitelj

Određuje koji su upravljački programi potrebni za podršku određenom uređaju i učitava te upravljačke programe

Upravitelj napajanja

Koordinira upravljanje napajanjem među uređajima

Izvršni dio Windows OS-a

Sigurnosni referentni monitor

Provodi pravila provjere valjanosti pristupa i kontrole

Upravitelj virtualne memorije

Upravlja virtualnim adresama, fizičkom memorijom i stranicama na disku i kontrolira hardverske i podatkovne strukture za upravljanje memorijom

Upravitelj procesa/dretvi

Stvara, upravlja i briše objekte procesa i dretvi

Upravitelj konfiguracije

Odgovoran za implementaciju i upravljanje registrom sustava, koji je spremište za postavke različitih parametara na razini cijelog sustava i po korisniku

Mogućnost naprednog poziva lokalne procedure (ALPC).

Implementira učinkovit međuprocetni mehanizam pozivanja procedura za komunikaciju između lokalnih procesa koji implementiraju usluge

Procesi korisničkog moda

- Windows podržava četiri osnovne vrste procesa u korisničkom načinu rada:

Posebni procesi sustava

- Usluge korisničkog načina rada potrebne za upravljanje sustavom

Servisni procesi

- Spooler pisača, zapisnik događaja i komponente korisničkog načina rada koje surađuju s upravljačkim programima uređaja i raznim mrežnim uslugama

Podsustavi okoline

- Omogućite različite osobnosti OS-a

Korisnički programi

- Izvršne datoteke (EXE) i DLL-ovi koji pružaju funkcionalnost koju korisnici pokreću kako bi iskoristili sustav

Model klijent/poslužitelj

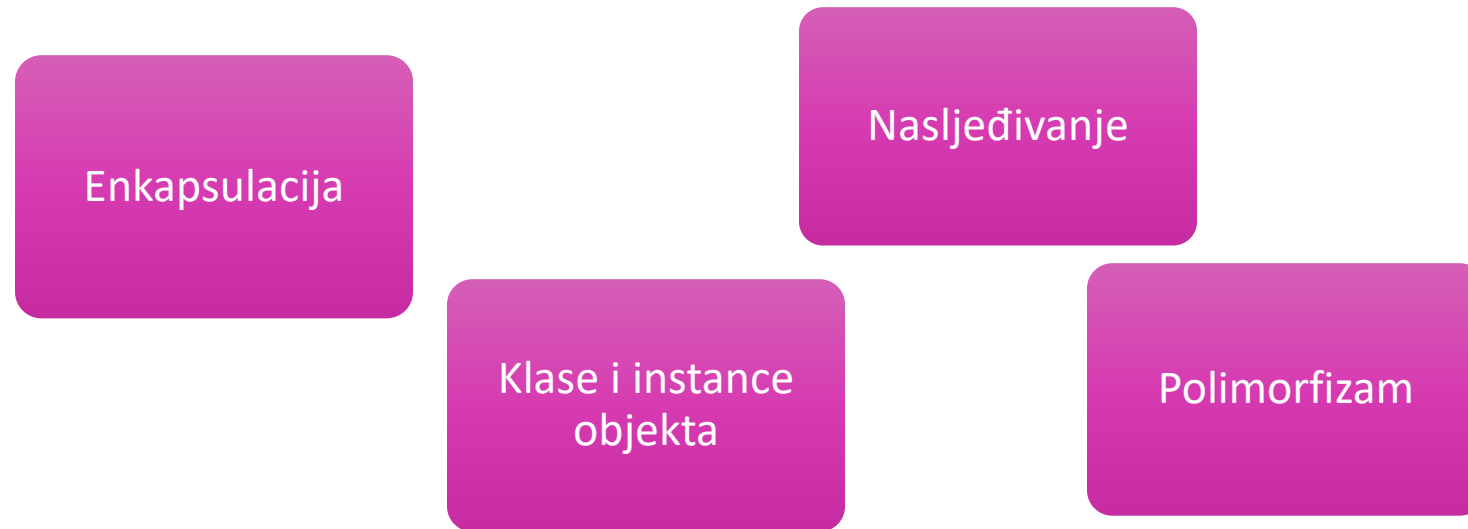
- Usluge Windows OS-a, podsustavi okoline i aplikacije strukturirani su pomoću modela klijent/poslužitelj
- Uobičajeno u distribuiranim sustavima, ali se može koristiti unutar jednog sustava
- Procesi komuniciraju putem RPC-a
- Prednosti:
 - Pojednostavljuje izvršavanje
 - Poboljšava pouzdanost
 - Omogućuje jedinstveno sredstvo za komunikaciju aplikacija s uslugama putem RPC-a bez ograničavanja fleksibilnosti
 - Pruža prikladnu osnovu za distribuirano računanje

Dretve i SMP

- Dvije važne karakteristike Windowsa su njegova podrška za dretve i za simetričnu višeprocesnu obradu (SMP)
 - OS se morež izvoditi na bilo kojem dostupnom procesoru, a različite rutine mogu se izvršavati istovremeno na različitim procesorima
 - Windows podržava korištenje više dretvi izvršavanja unutar jednog procesa.
 - Više dretvi unutar istog procesa može se istovremeno izvršavati na različitim procesorima
 - Procesi poslužitelja mogu koristiti više dretvi za obradu zahtjeva od više od jednog korisnika istovremeno
 - Windows pruža mehanizme za dijeljenje podataka i resursa između procesa i fleksibilne međuprocesne komunikacijske mogućnosti

Windows objekti

- Windows se uvelike oslanja na koncepte objektno orijentiranog dizajna
- Ključni objektno orijentirani koncepti koje koristi Windows su:



Tradicionalni UNIX sustavi

- Razvijen u Bell Labsu i postao operativan na PDP-7 1970. godine
- Prva značajna prekretnica bila je prijenos UNIX sustava s PDP-7 na PDP-11
 - Prvi je pokazao da će UNIX biti OS za sva računala
- Sljedeća prekretnica bilo je ponovno pisanje UNIX-a u programskom jeziku C
 - Prikazane su prednosti korištenja jezika visoke razine za sistemski kod
 - Prvi put je opisan u tehničkom časopisu 1974
 - Prva široko dostupna verzija izvan Bell Labsa bila je verzija 6 1976. godine
 - Verzija 7, objavljena 1978., predak je većine modernih UNIX sustava
 - Najvažniji od ne-AT&T sustava bio je UNIX BSD (Berkeley Software Distribution), koji je radio prvo na PDP, a zatim na VAX računalima

BSD

- Berkeley Software Distribution
- 4.x BSD se široko koristi u akademskim instalacijama i poslužio je kao osnova brojnih komercijalnih UNIX proizvoda
- 4.4 BSD je bila konačna verzija BSD-a koju je objavio Berkeley
- Postoji nekoliko široko korištenih verzija BSD-a otvorenog koda
 - FreeBSD
 - Popularno za internetske poslužitelje i vatrozide
 - Koristi se u brojnim ugrađenim sustavima
 - NetBSD
 - Dostupan za mnoge platforme
 - Često se koristi u ugrađenim sustavima
 - OpenBSD
 - OS otvorenog koda koji poseban naglasak stavlja na sigurnost

Solaris 11

- Oracleovo izdanje UNIX-a temeljeno na SVR4
- Pruža sve značajke SVR4 plus niz naprednijih značajki kao što su:
 - Jezgra s više dretvi koja se potpuno može isključiti
 - Potpuna podrška za SMP
 - Objektno orijentirano sučelje za datotečne sustave

LINUX

- Započeo je kao UNIX varijanta za IBM PC
- Linus Torvalds, finski student informatike, napisao je početnu verziju
- Linux je prvi put objavljen na Internetu 1991. godine
- Danas je to potpuno opremljen UNIX sustav koji radi na gotovo svim platformama
- Besplatan je i izvorni kod je dostupan
- Ključ uspjeha Linuxa bila je dostupnost besplatnih softverskih paketa pod pokroviteljstvom Free Software Foundation (FSF)
- Vrlo modularan i lako se konfigurira

Modularna struktura

- Razvoj Linuxa je globalan i obavlja ga slabo povezana skupina neovisnih programera
- Iako Linux ne koristi pristup mikrojezgri, on postiže mnoge od potencijalnih prednosti pristupa pomoću svoje posebne modularne arhitekture
- Linux je strukturiran kao zbirka modula, od kojih se određeni broj može automatski učitati i isprazniti na zahtjev

Učitavi moduli

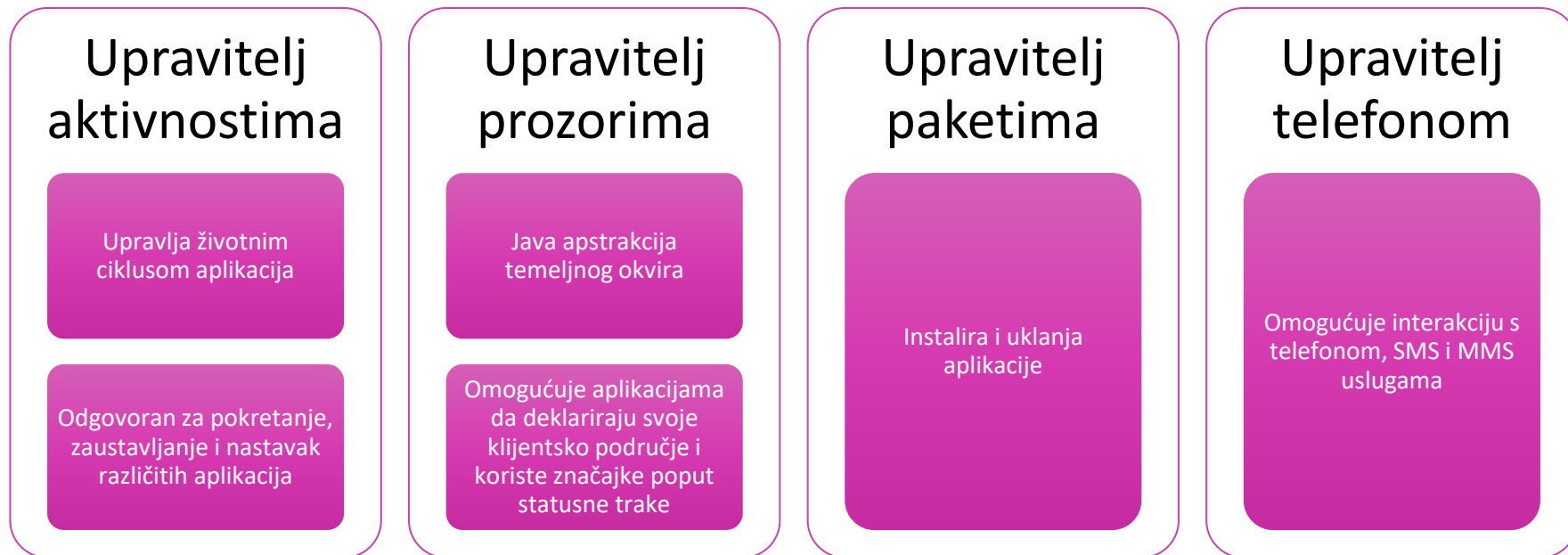
- Relativno neovisni blokovi
- Modul je objektna datoteka čiji se kod može povezati s kernelom i prekinuti vezu s njim tijekom vremena izvođenja
- Modul se izvršava u kernel modu u ime trenutnog procesa
- Imaju dvije važne karakteristike:
 - Dinamičko povezivanje
 - Mogućnost slaganja

Android OS

- Sustav temeljen na Linuxu izvorno dizajniran za mobilne telephone
- Najpopularniji mobilni OS
- Razvoj je napravio Android Inc., koji je Google kupio 2005. godine
- Prva komercijalna verzija (Android 1.0) objavljena je 2008
- Najnovija verzija je Android 12.0
- Android ima aktivnu zajednicu programera i entuzijasta koji koriste izvorni kod projekta Android Open-Source (AOSP) za razvoj i distribuciju vlastitih modificiranih verzija operativnog sustava
- Priroda Androida otvorenog koda bila je ključ njegova uspjeha

Programski okvir (Application Framework)

- Pruža građevne blokove visoke razine koji su dostupni putem standardiziranih API-ja koje programeri koriste za stvaranje novih aplikacija
 - Arhitektura je dizajnirana da pojednostavi ponovnu upotrebu komponenti
- Ključne komponente:



Programski okvir

Dobavljači sadržaja

Ove funkcije obuhvaćaju podatke aplikacije koje je potrebno dijeliti između aplikacija kao što su kontakti

Upravitelj resursima

Upravlja resursima aplikacije, kao što su bitmape

Sustav za prikaz

Pruža primitive korisničkog sučelja (UI) kao i događaje korisničkog sučelja

Upravitelj lokacijom

Omogućuje razvojnim programerima pristup uslugama temeljenim na lokaciji, bilo putem GPS-a, ID-ova mobilnih tornjeva ili lokalnih Wi-Fi baza podataka

Upravitelj obavjestima

Upravlja događajima, kao što su pristigle poruke i sastanci

XMPP

Omogućuje standardizirane funkcije razmjene poruka između aplikacija

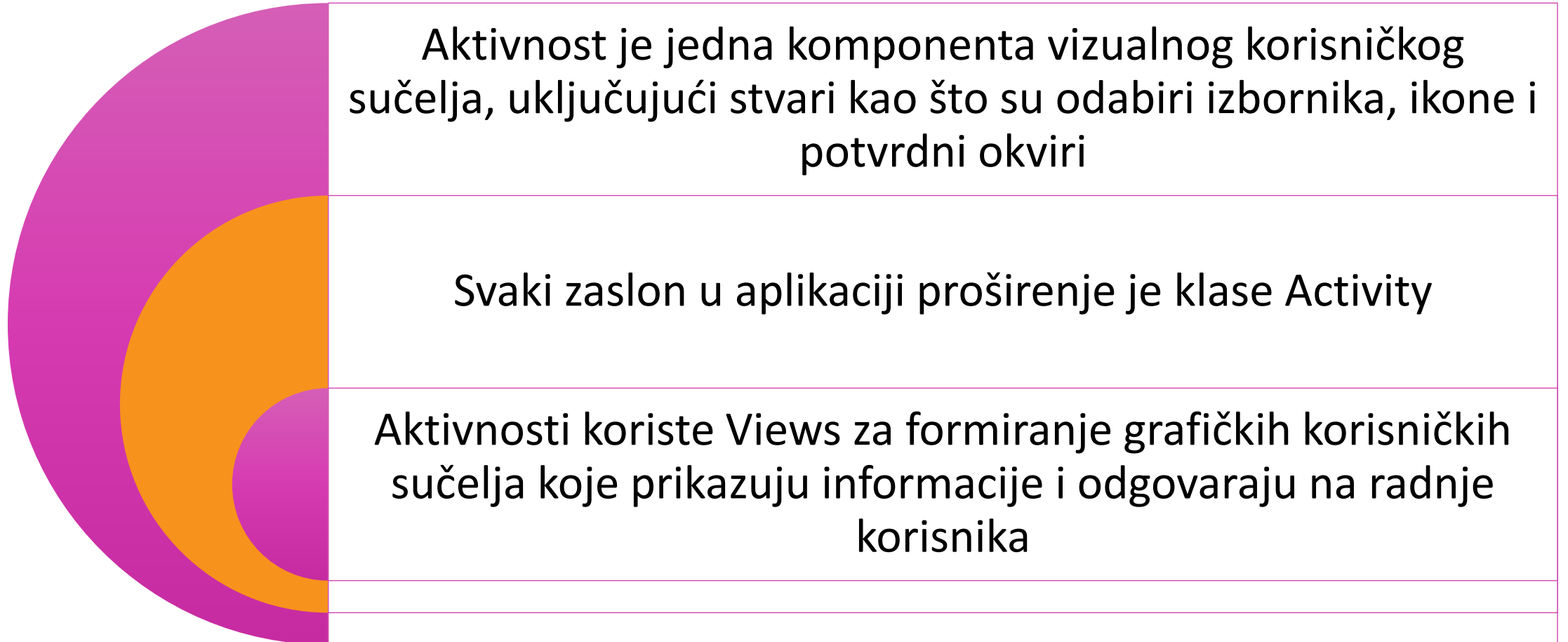
Biblioteke sustava

- Zbirka korisnih funkcija sustava napisanih na C ili C++ i koje koriste različite komponente Android sustava
- Poziva se iz aplikacijskog okvira i aplikacija putem Java sučelja
- Izložen razvojnim programerima kroz Android aplikacijski okvir
- Neke od ključnih biblioteka sustava uključuju:
 - Surface Manager
 - OpenGL
 - Media Framework
 - SQL Database
 - Browser Engine
 - Bionic LibC

Android Runtime

- Većina Android softvera se preslikava u format bajtkoda koji se zatim pretvara u izvorne upute na samom uređaju
- Ranija izdanja Androida koristila su shemu poznatu kao Dalvik, međutim Dalvik ima niz ograničenja u smislu skaliranja do većih memorija i višejezgrene arhitekture
- Novija izdanja Androida oslanjaju se na shemu poznatu kao Android runtime (ART)
- ART je u potpunosti kompatibilan s Dalvikovim postojećim formatom bajtkoda tako da programeri aplikacija ne moraju mijenjati da bi bili izvršni pod ART-om
- Svaka Android aplikacija radi u svom vlastitom procesu, s vlastitom instancom Dalvik VM-a

Aktivnosti



Upravljanje pšotrošnjom energije

Alarmi

- Implementirano u jezgri Linuxa i vidljivo je programeru aplikacije kroz AlarmManager u jezgri RunTime biblioteka
- Implementiran je u kernelu tako da se alarm može aktivirati čak i ako je sustav u stanju mirovanja
 - To omogućuje sustavu da pređe u stanje mirovanja, štedeći energiju, iako postoji proces koji zahtijeva buđenje

Wakelocks

- Sprječava Android sustav da uđe u stanje mirovanja
- Ova se zaključavanja zahtijevaju putem API-ja kad god aplikacija zahtijeva da jedan od upravljanih perifernih uređaja ostane uključen
- Aplikacija može zadržati jedno od sljedećih wakelockova:
 - Full_Wake_Lock
 - Partial_Wake_Lock
 - Screen_Dim_Wake_Lock
 - Screen_Bright_Wake_Lock

Sažetak

- Ciljevi i funkcije operacijskog sustava
 - Sučelje korisnika/računala
 - Voditelj resursa
- Evolucija operativnih sustava
 - Serijska obrada
 - Jednostavni/multiprogramirani/batch sustavi s dijeljenjem vremena
- Glavna postignuća
- Razvoj koji vodi do modernih operativnih sustava
- Tolerancija kvarova
 - Temeljni koncepti
 - Greške
 - OS mehanizmi
- Razmatranja o dizajnu OS-a za višeprocorske i višejezgrene sustave
- Pregled sustava Microsoft Windows
- Tradicionalni Unix sustavi
 - Povijest/opis
- Moderni Unix sustavi
 - BSD
 - Solaris 11
- Linux
 - Povijest
 - Modularna struktura
 - Komponente jezgre
- Android
 - Arhitektura softvera/sustava
 - Aktivnosti
 - Upravljanje napajanjem

An abstract graphic on the left side of the slide, composed of thick, curved lines. The lines are primarily pink and magenta, with a section at the bottom left transitioning into a bright orange color. The lines curve and loop, creating a dynamic, organic shape.

**Thank you for
your attention!**