

# 1

## Uvod u operativne sisteme

Računarski sistem je složen skup tehničkih uređaja (hardvera), programa koji zadaju instrukcije hardveru, i podataka koji se obrađuju. Ljudi koji koriste računare, bilo kao operateri, programeri ili projektanti, u radu se pridržavaju različitih formalnih i neformalnih procedura. Mnoge procedure se mogu pretočiti u programe, tako da ih obavljaju računari, a ne ljudi.

Prevodioci (engl. *compilers*) omogućavaju ljudima da programe pišu na jeziku koji razume čovek, i da ih zatim prevedu na mašinski jezik, tj. na jezik koji razume mašina. Povezivači (engl. *linkers*) i punioci (engl. *loaders*) omogućavaju sprezanje više mašinskih programa u jednu programsku celinu. Ta celina se može upotrebiti više puta i uvek se smešta na razne lokacije u memoriji. Pri tome, treba odgovoriti na sledeća pitanja:

- Kako se više programa može istovremeno smestiti u memoriju?
- Ko će izabrati koji će program biti u memoriji?
- Koji će program i kada koristiti procesor?
- Kako ćemo obezbediti da svaki program dobije resurse?

Operativni sistem sadrži odgovore na sva ova pitanja.

U ovom poglavlju je u kraćim crtama opisan istorijat operativnih sistema i navedene su njihove funkcije, karakteristike i poželjne osobine. Operativni sistemi su podeljeni na osnovu različitih kriterijuma i dat je kraći pregled njihove strukture. Detaljnije informacije o funkcijama pojedinih slojeva u hijerarhijskom modelu operativnog sistema nisu predmet izučavanja u ovom poglavlju.

---

---

### *Sadržaj poglavlja*

- 1.1 Istorijat operativnih sistema
  - 1.2 Definicija i funkcije operativnog sistema
  - 1.3 Karakteristike operativnih sistema
  - 1.4 Vrste operativnih sistema
  - 1.5 Opšti pregled strukture operativnih sistema
  - 1.6 Operativni sistemi UNIX i Linux
  - 1.7 Pitanja i zadaci
- 
-

## 1.1 ISTORIJAT OPERATIVNIH SISTEMA

Istorijat operativnih sistema posmatračemo prateći istorijat samih računara.

Računari prve generacije (1945–1955), čiju su osnovu činile vakuumske cevi (do 20.000 cevi po računaru), bili su ogromnih dimenzija i veoma skupi. Koristila ih je uglavnom vojska. Ovi računari su bili jako spori, programiralo se na mašinskom jeziku, dok su simbolički jezici (uključujući i assembler), kao i operativni sistemi, u to vreme bili nepoznati. Ljudi koji su radili na tim računarima obavljali su sve poslove – od programiranja do održavanja računara.

U prvoj generaciji računara, opsluživanje računarskog sistema bilo je potpuno prepušteno operateru, koji je morao da pripremi sve što je potrebno da se zadatak obrade može obaviti. Čovek je, dakle, imao punu kontrolu nad računarskim sistemom. Operater je bio u mogućnosti da sve potrebne radnje obavi na vreme, jer je sistem bio spor i izvršavao je samo jedan program, tj. obavljao samo jedan zadatak. Može se reći da je iskorišćenje računarskog sistema, tj. njegovih najvažnijih resursa – centralnog procesora i centralne memorije – bilo slabo. Najveći deo vremena trošio se na poslove operatera i ulazno-izlazne operacije, a mnogo manji deo na rad centralnog procesora. Iako je ovakav sistem bio krajnje neefikasan, odnos tih vremenskih perioda bio je u prihvatljivim granicama zbog relativno male brzine samog centralnog procesora.

Osnovu računara druge generacije (1955–1965) činili su tranzistori, pa su računari postali manji, pouzdaniji i jeftiniji. Računare druge generacije su, osim vojske, kupovale i velike korporacije i univerziteti. Računari su bili smešteni odvojeno, u posebnim sobama, koje su se delile u tri funkcionalne celine: ulazna soba, centralni računar i izlazna soba. Programeri su pisali programe na papiru, na programskom jeziku FORTRAN, zatim su se ti programi prenosili na bušene kartice, koje su se ostavljale u sobi sa ulaznim računarom (engl. *input room*). Operator sistema je, zatim, uzimao bušene kartice i ubacivao ih u računar, i to prvo kartice sa prevodiocem FORTRAN-a, a potom bušene kartice s programom koji treba izvršiti. Glavni računar je obavljao posao, a rezultat se dobijao takođe na bušenim karticama, koje su se prenosile u prostoriju s rezultatima (engl. *output room*). Ovde se mnogo vremena trošilo na šetanje između raznih prostora s bušenim karticama. Operativni sistem kao zaseban pojam još uvek nije postojao.

Dalje se kao poboljšanje uvodi paketna, tj. grupna obrada (engl. *batch processing*), zasnovana na upotrebi magnetne trake – uređaja mnogo bržeg od bušenih kartica. Pri paketnoj obradi, u ulaznoj sobi s poslovima sakuplja se jedna količina sličnih programa (na primer, svi programi koji zahtevaju prevodilac FORTRAN-a), koji se pomoću jeftinijeg računara (npr. IBM 1401) s bušenih kartica prenose na magnetnu traku. Posle toga se magnetna traka prenosi u sobu s glavnim računarom, tj. s moćnijim i skupljim računarom, predviđenim za izvršavanje programa (npr. IBM 7094). U glavni računar se učitava poseban program koji je zadužen da s trake sa poslovima programe redom učitava i izvršava. Taj program se može smatrati pretkom operativnih sistema. Nakon izvršavanja programa, rezultati se snimaju na drugu magnetnu traku koju operater prenosi do trećeg računara, zaduženog za prebacivanje rezultata s magnetne trake na bušene kartice. Manji računari (ulazno-izlazni) nisu direktno vezani za glavni računar, što znači da rade u *off-line* režimu.

U drugoj generaciji računarskih sistema povećava se brzina rada centralnog procesora, kapaciteti centralne memorije i eksternih memorija, pojavljuju se nove i brže ulazno-izlazne jedinice. Programi se pišu na simboličkom mašinskom jeziku, assembleru ili na višem programskom jeziku (FORTRAN). Operater više nije u stanju da efikasno opslužuje računarski sistem, jer su njegove reakcije suviše spore. Jedino rešenje se moglo naći u prebacivanju niza kontrolnih funkcija sa operatera na sam računarski sistem, to jest na posebne kontrolne programe. Otuda su funkcije opsluživanja i upravljanja sistemom bile podeljene između operatera i kontrolnih programa. Ti programi se uključuju u određenim situacijama, kao što su, na primer, priprema programa za izvođenje, kontrola ulaznih i izlaznih uređaja i razni poslovi oko učitavanja programa i pripreme za njegovo izvođenje. Dakle, u računarima druge generacije razlikuju se dve osnovne vrste programa: kontrolni i korisnički.

Računari treće generacije (1965–1980) prave se od integrisanih kola (IC). Početkom šezdesetih većina proizvođača pravi dve vrste računara: jednu bržu verziju (kao IBM 7094) i jednu slabiju (kao IBM 1401), što je skup poduhvat. Novi korisnici računara najpre žele slabije modele koji su jeftiniji, dok će im jači, brži i skuplji modeli biti potrebni tek nakon izvesnog vremena. IBM taj problem pokušava da rešeri uvođenjem klase računara IBM System/360. To je serija kompatibilnih računara različitih snaga. Svaki od ovih računara je pogodan i za naučnu i za poslovnu primenu, pa je time podela računara na ove dve vrste nestala. Ovaj koncept su preuzeli i ostali proizvođači računara. Računari iz serije System/360 radili su pod operativnim sistemom OS/360, koji je bio veoma glomazan i prepun grešaka.

S razvojem discipline poznate pod imenom softversko inženjerstvo (engl. *software engineering*), uvode se nove funkcije:

- multiprogramiranje, (engl. *multiprogramming*);
- višestruke ulazno-izlazne (U/I) operacije (engl. *spool*);
- podela računarskog vremena (engl. *time-sharing*).

Kada neki program čeka na rezultate ulazno-izlaznih operacija, procesor je neiskorišćen, pa se gubi procesorsko vreme. Ovaj problem nije toliko izražen kod programa koji retko zahtevaju ulazno-izlazne operacije (na primer, naučno orijentisani programi), ali jeste kod poslovnih programa. Multiprogramiranje je tehnika kojom se postiže bolje iskorišćavanje procesora: memorija se deli na particije u koje se učitavaju različiti programi, to jest poslovi (engl. *jobs*). Dok neki program čeka na ulazno-izlaznu operaciju, procesor može izvršavati drugi program. Na taj način, ako imamo dovoljan broj programa u memoriji, procesor se stalno upotrebljava.

Spuling (engl. *Spool – Simultaneous Peripheral Operation On Line*) jeste tehnika koja omogućava da se nedovoljna brzina ulazno-izlaznih uređaja kompenzuje upotrebom brzih uređaja kao što su trake, a naročito diskovi. Na taj način se omogućava istovremeno izvršenje više ulazno-izlaznih operacija. Brzi uređaj prihvata sve sa ulaza, a zatim se ulaz ka procesoru realizuje sa brzog uređaja. Ulaz se realizuje prebacivanjem sadržaja bušenih kartica na disk (traku) pomoću posebnog uređaja, a bez korišćenja procesora. To znači da se disk paralelno puni novim poslovima dok procesor

izvršava programe u memoriji. Kada jedan program završi rad, procesor na njegovo mesto može učitati drugi program sa diska. Slično, sve što se šalje na izlaz, prvo se prenosi na brzi uređaj, a zatim s njega na spore periferne uređaje.

Podela vremena (engl. *time-sharing*) jeste tehnika koja omogućava da svaki korisnik radi s računarem interaktivno, i to preko posebnog terminala koji je istovremeno i ulazni i izlazni uređaj za korisnika. Podela vremena je poseban oblik multiprogramiranja, gde svakom terminalu pripada dodeljeno procesorsko vreme. Posle isteka vremenskog kvantuma, tj. dodeljene količine procesorskog vremena, procesor se dodeljuje drugom terminalu. Ukoliko je terminal blokiran zbog čekanja na ulazno-izlazne operacije, procesor se i pre isteka kvantuma dodeljuje drugom terminalu.

Kod treće generacije računara posebno treba istaći pojavu dva operativna sistema – MULTICS i UNIX. Projekat MULTICS (*MULTIplexed Information and Computing Service*) neuspela je ideja kompanija MIT, Bell Labs i General Electric da se napravi moćan računar i operativni sistem koji će biti u stanju da radi s velikim brojem terminala. Osnovna ideja je preuzeta iz modela distribucije električne energije – u tom modelu, dovoljno je da svaki korisnik koji poželi da upotrebi neki električni aparat, taj isti aparat samo priključi na električnu mrežu. Sličan model su pokušali da naprave i sa računarima: ideja je da u jednom gradu postoji moćan centralni računar, a da građani kod kuće imaju terminale kojima preko modema pristupaju glavnom računaru. Ovaj model se može smatrati pretečom računarskih mreža i Interneta. Drugi operativni sistem, UNIX, uprošćena je varijanta MULTICS sistema, koja je doživela praktičnu realizaciju i ekspanziju do današnjih dana. Ken Thompson, jedan od naučnika i programera kompanije Bell Labs, koji je radio na razvoju projekta MULTICS, napisao je za računar PDP-7 mini verziju MULTICS sistema. Posle toga je nastao UNIX (UNI = jedan, X = CS = *Computing Service*).

Računari treće generacije zvali su se mini računari: prvi računar je DEC-ov (Digital Equipment Corporation) PDP-1, do tada najmanji i najjeftiniji računar. Koštao je tada „samo“ 120.000 dolara.

U trećoj generaciji računarskih sistema, zbog pojave multiprogramiranja i porasta brzina, veličine memorije i broja ulaznih i izlaznih jedinica, još više kontrolno-upravljačkih funkcija prebacuje se sa čoveka na računar. Dakle, čovek definitivno gubi mogućnost kontrole interne situacije u računarskom sistemu i upravljanja njome i sve što je moguće prebacuje se na računarski sistem, tj. na pojedine sistemске programe. Skup svih tih programa naziva se jednim imenom: operativni sistem. Programer se oslobađa niza složenih rutinskih poslova i pruža mu se mogućnost većeg angažovanja na kreativnom delu posla. No, pored kontrolno-upravljačkih programa u, računarima treće generacije razvijen je i čitav niz uslužnih programa, čiji je zadatak da dalje olakšaju i pojednostave upotrebu računarskih sistema. Zbog toga, prema nameni, softver možemo podeliti na sistemski i korisnički (aplikativni).

U četvrtoj generaciji računara (1980–1990), prvi put se pojavljuju personalni računari. Razvoj personalnih računara započinje pojavom LSI čipova, to jest čipova visokog stepena integracije (engl. *Large Scale Integration*). Računari su bili dovoljno jeftini, tako da su ih mogli priuštiti i više odseka iste firme ili univerziteta, dok su

personalni računari postali dovoljno jeftini da ih mogu imati i pojedinci. Poznatiji personalni računari su Spectrum, Commodore, Atari, zatim IBM PC, Apple Macintosh itd. U prve operativne sisteme za personalne računare spadaju MS-DOS i UNIX.

Paralelno s razvijanjem korisničkog softvera, razvija se i korisnički interfejs programa, to jest korisničko okruženje. Na taj način se osobama koje računarski sistem i same programe ne poznaju detaljno, omogućava da te programe uspešno koriste.

Pored klasičnih operativnih sistema javljaju se i dve nove vrste, a to su mrežni operativni sistemi i distribuirani operativni sistemi.

Mrežne operativne sisteme karakterišu računari povezani u mrežu. Ovi računari zadržavaju relativno visok stepen autonomije – svaki računar ima svoj operativni sistem – a u mogućnosti su da međusobno razmenjuju podatke pomoću odgovarajućih protokola. Operativni sistemi mogu biti različiti, potreban je samo zajednički protokol, tj. zajednički jezik za komunikaciju. Korisnik jednog računara može se prijaviti na drugi, preuzeti neke datoteke itd. Korisnik zna da nije sam u mreži, tj. svestan je različitih računara s kojima komunicira preko mreže.

Distribuirani operativni sistemi su mnogo ozbiljnija varijanta u mrežnom okruženju, zato što osim deljenja i migracije datoteka i štampača omogućavaju i deljenje procesa, tj. programa. Korisnici ovaj sistem vide kao jednoprocesorski sistem, ali se, u stvari, radi o operativnom sistemu namenjenom za rad s više procesora koji su fleksibilno povezani preko mreže. To znači da postoji više računara povezanih u mrežu, ali samo jedan operativni sistem, upravlja svim resursima u mreži. U pravom distribuiranom sistemu, korisnik ne treba da vodi računa o tome gde su smeštene njegove datoteke ili gde se izvršava njegov program – to je posao distribuiranog operativnog sistema. Distribuirani operativni sistem se, dakle, ponaša kao jedinstvena celina. Korisnik ne mora znati da je umrežen s drugim računarima – on ceo sistem vidi kao jedan računar.

## 1.2 DEFINICIJA I FUNKCIJE OPERATIVNOG SISTEMA

Operativni sistem objedinjuje raznorodne delove računara u skladnu celinu i sakriva od korisnika detalje funkcionisanja ovih delova koji nisu bitni za korišćenje računara. Operativni sistem radi sledeće:

- upravlja programima, podacima i delovima od kojih se računar sastoji (procesor, kontroleri, radna memorija), s ciljem da oni budu što celishodnije upotrebljeni;
- obezbeđuje pristupačno radno okruženje za krajnjeg korisnika računara, tako što računar kao mašinu koja rukuje bitovima, bajtovima i blokovima pretvara u mašinu koja rukuje datotekama i procesima.

U opštem smislu, operativni sistem se može definisati kao skup programa koji upravljaju resursima računarskog sistema i obezbeđuju interfejs ka korisniku.

Prva funkcija operativnog sistema je upravljanje resursima računara (engl. *resource management*). Pod pojmom resurs podrazumevamo sve što je programu potrebno za rad. Resursi mogu biti hardverski (procesor, memorija, ulazno-izlazni uređaji) i

softverski (programi, podaci, tj. datoteke svih vrsta). Zadatak operativnog sistema je da vodi računa o resursima računara, to jest da zadovolji potrebe programa, da prati koji program koristi koje resurse itd. Na primer, ukoliko dva korisnika višekorisničkog sistema istovremeno žele nešto da štampaju, operativni sistem je dužan da obezbedi dostupnost štampača programima tih korisnika i da spreči mešanje podataka poslatih na štampu.

Kako je pri takvoj arhitekturi računara i hardvera većini korisnika programiranje krajnje komplikovano, operativni sistem obezbeđuje i takozvanu virtuelnu mašinu, to jest jedinstven pogled na računarski sistem s tačke gledišta korisnika, nezavisan od konfiguracije računara, konkretnog hardvera i same arhitekture. Ova nezavisnost može biti izražena u većoj ili manjoj meri. Kao primer uzmimo kontroler za disketni uređaj čije osnovne komande za čitanje podataka sa diskete i upisivanje podataka na nju zahtevaju da se prilikom njihovog pozivanja navede veliki broj parametara. Prilikom pristupa disketnom uređaju treba voditi računa o tome da li je motor uključen, zatim treba naći odgovarajuću stazu, pa sektor itd. To bi trebalo uraditi svaki put kada želimo nešto da upišemo na disketu ili da čitamo sa nje. Zadatak operativnog sistema kao virtuelne mašine jeste da te stvari radi umesto nas i da nam pruža neke funkcije višeg nivoa apstrakcije radi lakšeg pristupa hardveru.

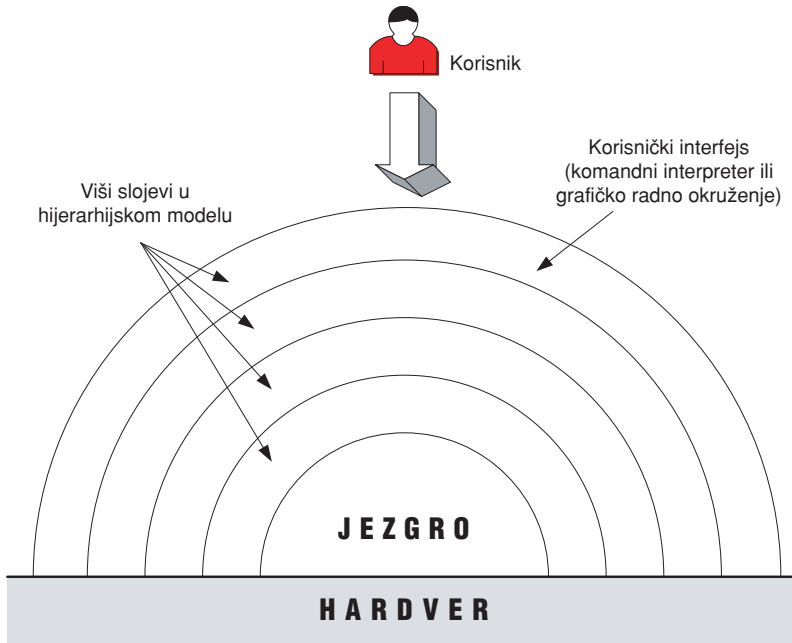
Ukratko rečeno, operativni sistem je skup sistemskih programa koji posreduje između korisnika računara i računarskog hardvera, a cilj mu je da:

- izvršava korisničke programe i olakša rešavanje korisničkih problema;
- korišćenje računarskog sistema učini podesnijim za korisnika;
- omogućiti što efikasnije iskorišćenje računarskog hardvera.

Kao takav, operativni sistem je jedan od najvažnijih i najsloženijih delova računarskog sistema i sastoji se od više relativno nezavisnih celina. Treba imati na umu da svaki proizvođač računara ima svoje operativne sisteme, pa je teško dati opštu strukturu operativnog sistema. Na slici 1.1 prikazan je hijerarhijski model operativnog sistema.

Pojedini delovi predstavljaju nivoe operativnog sistema. Hijerarhijski model ima sledeći smisao: na posmatranom nivou operativnog sistema mogu se zahtevati usluge samo od njegovih nižih nivoa, a nikako od viših. Najniži sloj je poznat kao jezgro operativnog sistema (engl. *nucleus*, *kernel*).

Ne postoje čvrsto definisana pravila koja regulišu raspodelu funkcija operativnog sistema po nivoima. Zbog odnosa veličine operativnog sistema i radne memorije, većina operativnih sistema ne može da stane u radnu memoriju. Zato se u memoriji uvek nalaze samo najvažniji delovi operativnog sistema, takozvani rezidentni delovi, koji aktiviraju i završavaju korisničke programe, dodeljuju memoriju i datoteke, i obavljaju ulazno-izlazno operacije. Rezidentni deo operativnog sistema mora obavezno podržavati mehanizam prekida, jer je on osnova višeprogramskog rada i komuniciranja računara sa spoljnim svetom. Deo operativnog sistema koji mora uvek postojati u radnoj memoriji obično se naziva jezgro ili nukleus. Funkcije koje koriste svi nivoi moraju se smestiti u jezgro operativnog sistema. Ostali delovi se ubacuju u memoriju kada su potrebni i izbacuju kada više nisu potrebni.



Slika 1.1 Hijerarhijski model operativnog sistema.

## Funkcije operativnog sistema

Funkcije koje treba da obavlja operativni sistem mogu se izvesti na osnovu očekivanih funkcija jednog računarskog sistema, kao što su:

[1] Automatsko funkcionisanje računarskog sistema

Operativni sistem mora da obezbedi funkcionisanje računarskog sistema bez intervencije operatera, iz veoma prostog razloga – ljudske intervencije su mnogo sporije od računara. U aktivnosti koje operativni sistem treba da izvršava bez ljudske intervencije spadaju, na primer, punjenje memorije programom iz sekundarnih memorija i izvršavanje jednog ili više programa.

[2] Mogućnost planiranja i raspoređivanja poslova, i postojanje jezika za upravljanje poslovima

Pod planiranjem i raspoređivanjem poslova (engl. *scheduling*) podrazumeva se određivanje koji će se posao koji je spreman za rad izvršavati, tj. kome će se dodeliti glavni procesor. Radi kontrole automatske sekvence, operativni sistem uvodi kontrolne naredbe, koje služe za upravljanje radom celog računarskog sistema. Preciznije, uvodi se jezik za upravljanje poslovima (engl. *job control language*), koji se interpretira preko operativnog sistema.

[3] Multiprogramiranje

Multiprogramiranje je tehnika za pokretanje više programa na istom računaru istovremeno, tako što svaki dobije deo memorije, a procesor se dobija prema funkciji za raspoređivanje poslova.

## [4] Eliminisanje zavisnosti U/I operacija

Kako su U/I operacije mnogo sporije od procesora, operativni sistem mora da izoluje U/I operacije od procesora, što se postiže upotrebom brzog medijuma za privremeno memorisanje svih U/I podataka. Taj uređaj je u prošlosti bila traka, a danas je to po pravilu disk. U cilju efikasnosti, U/I operacije po mogućstvu treba što više preklapati ili kombinovati s drugim procesorskim poslovima. U tom kontekstu, realizovane su dve hardverske strukture: kanal (engl. *channel*) i tehnika prekida (engl. *interrupt*). Pod kanalom se podrazumeva uređaj koji kontroliše jedan ili više perifernih uređaja, a sposoban je da prenosi podatke između perifernih uređaja i memorije bez intervencije glavnog procesora. Kanal po pravilu za takve prenose koristi usluge kvalitetne hardverske komponente koja se zove DMA kontroler (engl. *Direct Memory Access*). Tehnika prekida je univerzalna metoda za obavestavanje operativnog sistema da se neka akcija, to jest U/I komanda, završila. Tehnika prekida obezbeđuje visoke performanse i izolovanost od U/I uređaja: DMA prenosi podatke po kanalu, a odgovarajućim prekidnim signalom sistem se obavestava da je transfer završen. Prekidnim signalom se upravljanje centralnim procesorom prenosi na neku drugu lokaciju. Pri tom se čuva prethodna vrednost programskog brojača, što omogućava da se prekinuti program nastavi.

Dakle, na osnovu definicije operativnog sistema i očekivane funkcionalnosti računarskog sistema, zaključuje se da operativni sistem mora obavljati sledeće funkcije:

- upravljanje poslovima (sekvenciranje i raspoređivanje poslova) i interpretacija komandnog jezika;
- rukovanje ulazno-izlaznim operacijama;
- rukovanje greškama i prekidima;
- upravljanje resursima;
- omogućavanje višestrukog pristupa;
- zaštita resursa od zlonamernih napada, slučajnih grešaka korisnika i grešaka u korisničkim programima i samom operativnom sistemu;
- obezbeđivanje dobrog interfejsa za operatora i korisnika;
- obračun korišćenja računarskih resursa.

### 1.3 KARAKTERISTIKE OPERATIVNIH SISTEMA

Karakteristike operativnih sistema su:

## [1] Konkurentnost

Konkurentnost (engl. *concurrency*) je postojanje više simultanih, paralelnih aktivnosti. Primeri su preklapanje U/I operacija i operacija izračunavanja ili koegzistencija više programa u memoriji. Konkurentnost izaziva probleme pri prelasku s jedne aktivnosti na drugu, probleme zaštite jedne aktivnosti od druge i sinhronizacije aktivnosti koje su međusobno zavisne.



## [2] Deoba resursa

Konkurentne aktivnosti mogu da zahtevaju deljenje (engl. *sharing*) resursa ili informacija. Razlozi za deljenje resursa su višestruki. Pre svega, to je ukupna cena, jer je obezbeđivanje dovoljnih resursa za svakog korisnika najčešće preskupo. Poželjno je nadovezivanje rada jednog korisnika na rad drugog korisnika, deljenje podataka, tj. korišćenje istih podataka između različitih programa, i otklanjanje redundanse resursa.

## [3] Postojanje dugotrajne memorije

Potreba za deljenjem programa i podataka implicira potrebu za trajnim skladištenjem podataka s mogućnošću brzog pristupa (engl. *long-term storage*). To omogućavaju uređaji velikog kapaciteta, tj. sekundarne memorije, koji su uglavnom magnetni. Pri tome treba rešiti sledeće probleme: obezbeđenje jednostavnog pristupa podacima, zaštita od štetnih uticaja bilo kakve vrste (zlonamernih ili nehotaćnih) i zaštita od grešaka samog sistema.

## [4] Nedeterminizam

Operativni sistem mora biti deterministički orijentisan – znači, kad izvršava isti program sa istim podacima, mora da daje isti rezultat, bez obzira na to da li će to raditi danas, sutra, ili za mesec dana. Na drugoj strani, operativni sistem mora karakterisati nedeterminističko ponašanje (engl. *nondeterminancy*), što znači da mora da odgovori na masu zahteva i događaja koji se mogu desiti na nepredvidiv način (zahtevi za resursima, greške u toku izvršavanja programa, prekidni signali perifernih uređaja). Operativni sistem ne može predvideti sve situacije koje mogu nastupiti i mora biti spreman za sve moguće sekvence događaja.

## Poželjne osobine operativnog sistema

U poželjne osobine operativnog sistema spadaju:

## [1] Visok nivo efikasnosti

Efikasnost operativnog sistema može se precizno predstaviti samo pomoću više kriterijuma, pri čemu značaj svakog kriterijuma u globalnoj predstavi zavisi od vrste i namene operativnog sistema. Kriterijumi su sledeći:

- srednje vreme između poslova;
- vreme neiskorišćenosti operativnog sistema;
- vreme prolaska (engl. *turn-around time*) za paketnu obradu;
- vreme odziva (engl. *response time*) za interaktivne sisteme;
- iskorišćenost resursa (engl. *resource utilisation*);
- propusna moć (broj poslova po satu, to jest ukupna veličina posla koji može biti obavljen interaktivno u nekom vremenu).

Uvešćemo na ovom mestu merilo efikasnosti (engl. *efficiency*),  $e$ , kao odnos vremena kada procesor radi korisne stvari i ukupnog vremena za koje se taj posao obavi:  $e = t_{\text{korisno}} / t_{\text{ukupno}}$ . Pri tome se  $e$  mora nalaziti u intervalu od 0 do 1 ( $0 < e < 1$ ).

Uvešćemo, takođe, merilo gubitka vremena,  $o$  (engl. *overhead*), kao odnos vremena kada procesor radi poslove vezane za održavanje samog sistema (engl. *housekeeping jobs*) i ukupnog vremena:  $o = t_{\text{održavanje}} / t_{\text{ukupno}}$ . Takođe važi  $0 < o < 1$  i  $e + o = 1$ .

[2] Visok nivo pouzdanosti

Idealno, operativni sistem bi trebalo da bude potpuno bez grešaka, ali to u realnosti nije moguće. Merilo pouzdanosti je broj grešaka, tj. srednje vreme između dve greške ili dva otkaza.

[3] Jednostavnost održavanja

Treba omogućiti da sistem održava što manje ljudi. Da bi se to postiglo, potrebna je modularna struktura, sa jasno definisanim interfejsima među modulima i, naravno, s dobrom pratećem dokumentacijom.

[4] Prihvatljiva veličina

Operativni sistem treba da zauzme što manje prostora u memoriji, mada – s povećanjem kapaciteta savremenih memorijskih sistema – to gubi važnost. Na drugoj strani, veliki sistemi su podložni greškama, a i njihov razvoj traje duže.

Ciljevi koji se postavljaju pri projektovanju operativnog sistema mogu se podeliti na systemske i korisničke. Što se korisničkih ciljeva tiče, operativni sistem treba da bude podesan za korišćenje, pouzdan, bezbedan i brz, i da se lako uči. Što se sistemskih ciljeva tiče, trebalo bi da bude takav da ga je jednostavno projektovati, implementirati, održavati, da je fleksibilan, pouzdan, bez grešaka i efikasan.

U prošlosti, operativni sistemi su pisani na asemblerskom jeziku, što je imalo svoje prednosti i mane. Prednosti su svakako u kompaktnosti izvršnog koda, koji je minimalan, a po performansama optimalan. Mane se ogledaju u tome što realizacija koda zahteva mnogo programerskog vremena. Zahvaljujući jeziku C i UNIX sistemu, operativni sistemi se danas pišu na višim programskim jezicima.

Implementacija u višem programskom jeziku donosi mnoge prednosti:

- izvorni kôd se mnogo brže piše;
- izvorni kôd je mnogo kompaktniji;
- kôd se mnogo lakše razume i lakše se otklanjaju greške (engl. *debug*);
- operativni sistem napisan na višem programskom jeziku mnogo se lakše prenosi na drugu računarsku arhitekturu, tj. na drugu vrstu procesora.

## 1.4 VRSTE OPERATIVNIH SISTEMA

Postoje brojne podele operativnih sistema na osnovu različitih kriterijuma: prema broju korisnika i/ili procesa, prema načinu obrade poslova, prema distribuciji procesorske snage i ostalih resursa, prema nameni i funkcionalnim osobinama.

## Klasifikacija prema broju korisnika i procesa

Prema broju korisnika, operativni sistemi se dele na jednokorisničke (engl. *single-user*) i višekorisničke (engl. *multiuser*).

Jednokorisnički sistemi, kao što im samo ime govori, obezbeđuju virtuelnu mašinu za samo jednog korisnika. To su ili računarski sistemi prilagođeni za jednu funkciju ili je reč o slabijim i jeftinijim konfiguracijama tipa mikroračunara. Uglavnom ih karakteriše jeftin hardver, solidna prateća programska podrška, jednostavan sistem datoteka, jednostavan U/I sistem, a u prošlosti i relativno slabe performanse. Međutim, prilike su se izmenile, hardver mikroračunara je ostao jeftin, ali su se performanse drastično poboljšale, što je dovelo do realizacije vrlo kvalitetnih operativnih sistema koji su i dalje jednokorisnički ali višeproceni, tj. sposobni da obavljaju više simultanih aktivnosti.

Višekorisnički sistemi su kvalitetni operativni sistemi koji zahtevaju jače hardverske konfiguracije. Tipičan višekorisnički sistem je UNIX, koji obezbeđuje simultani pristup za više korisnika istovremeno, pri čemu korisnici pristupaju sistemu preko posebnih terminala.

Prema broju simultanih aktivnosti, tj. prema broju procesa koji se mogu izvršavati paralelno ili kvaziparalelno, operativni sistemi se dele na jednoprocene (engl. *single-tasking*, *singleprocess*) i višeprocene (engl. *multitasking*, *multiprocess*).

Na osnovu kombinovanog kriterijuma mogu se izdvojiti tri vrste operativnih sistema:

- jednokorisnički jednoproceni (engl. *single-user*, *singletasking*), kao što je MS-DOS;
- jednokorisnički višeproceni (engl. *single-user*, *multitasking*), kao što su OS/2 i MS Windows 3.1/ 9x/ME;
- višekorisnički višeproceni (engl. *multiuser*, *multitasking*), kao što je UNIX, ali uslovno se mogu prihvatiti i MS Windows 2000/XP/2003 ukoliko obezbeđuju terminalske usluge (engl. *terminal services*).

## Klasifikacija prema nameni i načinu obrade poslova

Prema nameni, operativni sistemi se dele na operativne sisteme opšte namene (engl. *general purpose systems*), koji mogu da obavljaju razne poslove, kao što su obrada teksta i slike, i operativne sisteme specijalne namene, koji, po pravilu, služe za upravljanje procesima.

Prema načinu obrade poslova, operativni sistemi se klasifikuju kao:

- [1] Sistemi sa grupnom obradom (engl. *batch*)

Grupna (serijska, paketna) obrada je takav način rada računara u kome korisnici predaju svoje poslove na izvršenje posredstvom ulaznih jedinica, i koji se zatim odvijaju jedan za drugim u nizu, pri čemu korisnik nema mogućnost komuniciranja sa svojim poslom.

[2] Interaktivni sistemi (engl. *interactive systems*)

Interaktivne sisteme (nazivaju se još i *time-sharing* sistemi) karakteriše postojanje terminala za svakog korisnika, preko kojih korisnici zadaju poslove i komuniciraju sa svojim poslovima. Paralelnost u radu se postiže tako što se svakom korisničkom programu dodeljuje jedan kvantum vremena centralnog procesora, pa se na svaku poruku korisnika odaziva u roku od nekoliko sekundi. Po isteku vremenskog kvantuma dodeljenog jednom procesu, on se prekida, bilo da je završio s radom ili nije, a procesor se dodeljuje sledećem procesu u redu čekanja.

## [3] Kombinovani sistemi

Kombinovane sisteme karakteriše mogućnost istovremenog obavljanja interaktivnih poslova i paketne obrade. Na primer, korisnik može u pozadini pokrenuti nekoliko vremenski zahtevnih poslova koji ne zahtevaju interakciju sa njim; dok čeka na njihovo izvršenje, može čitati elektronsku poštu ili Web stranice.

## Klasifikacija prema funkcionalnim osobinama računarskog sistema

Prema funkcionalnim osobinama računarskog sistema za koji su namenjeni, operativni sistemi se dele u sledeće kategorije:

## [1] Operativni sistemi za velike računarske sisteme

Na velikim računarskim sistemima (engl. *mainframe systems*) prvi put su se pokretale mnoge komercijalne i naučne aplikacije. U svoje vreme bile su to fizički ogromne mašine kojima se upravljalo preko konzole. Zajednički ulazni uređaji bili su čitači bušenih kartica i magnetne trake, a izlazni uređaji, takođe, bušene kartice, trake ili štampači. Da bi se ubrzao rad sistema kod koga je bilo važno da priprema posla traje što kraće, vreme pripreme je skraćivano tako što su se slični poslovi udruživali u grupe i na taj način zajednički obrađivali. Uvodi se tehnika multiprogramiranja, a s njom i automatsko sekvenciranje poslova, koje bez operatera prebacuje kontrolu sa jednog posla na drugi. Da bi operativni sistem mogao da funkcioniše, njegov fundamentalni deo mora uvek biti u memoriji (engl. *resident monitor*); taj deo bira posao koji će se trenutno izvršavati. Po završetku posla ili nakon blokade zbog čekanja na ulazno-izlazne operacije, operativni sistem opet preuzima upravljanje i dodeljuje kontrolu drugom poslu.

## [2] Operativni sistemi za sisteme sa deljenjem vremena

Sistemi sa deljenjem vremena (engl. *time-sharing systems*) nastali su kao specijalna klasa velikih sistema u kojoj je svakom korisniku omogućena *on-line* komunikacija sa svojim poslom i operativnim sistemom. Vreme odziva se uvodi kao nov kriterijum za ocenu efikasnosti ove klase sistema. Opet imamo tehniku multiprogramiranja – što znači više poslova koji se u memoriji nalaze istovremeno – i tehniku deljenja vremena, gde posle isteka vremenskog kvantuma svaki posao mora da preda kontrolu drugom poslu i sačeka svoje novo vreme.

Ove sisteme podelićemo na dve posebne klase: upitni sistemi (engl. *file interrogation systems*) i transakcioni sistemi (engl. *transaction processing systems*).

Kod upitnih sistema, kao što su razni informacioni sistemi koji rade s velikim bazama podataka, veoma je bitno da vreme odziva za svaki postavljeni upit bude veoma kratko. Obično postoji dodatni sloj na vrhu operativnog sistema za upravljanje bazom podataka. Transakcioni sistemi su relativno slični upitnim, s tim što se kod njih često (na primer, nekoliko puta u sekundi) modifikuju baze podataka. Primer su sistemi za rezervaciju karata i bankarsko poslovanje. Ažurnost baze podataka je za transakcione sisteme izuzetno značajna.

[3] Operativni sistemi za stone računare

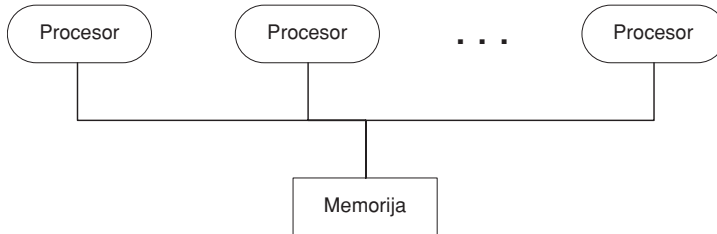
Stoni računari (engl. *desktop systems*) jesu mali računarski sistemi namenjeni jednom korisniku i dostupni po pristupačnim cenama. Prve verzije stonih računara imale su skromne performanse i hardver (na primer, malu količinu memorije i diskove manjih kapaciteta), ali se ta situacija drastično promenila. Osim poboljšanja performansi, kod ovih sistema je – razvojem pogodnih grafičkih korisničkih interfejsa – ostvarena ergonomičnost, to jest prilagođenost korisniku. Za razliku od velikih računarskih sistema, ova klasa je veoma fleksibilna i prilagodljiva novim tehnologijama, novim procesorima, novim memorijama, novim multimedijalnim i drugim perifernim uređajima. Brojna rešenja namenjena velikim sistemima primenjuju se u ovoj klasi računarskih sistema (na primer, tehnike keširanja). Ovu klasu karakteriše realizacija više vrsta operativnih sistema (DOS, MS Windows, Novell NetWare, UNIX, Linux).

[4] Operativni sistemi za višeprocessorske sisteme

Višeprocessorski sistemi (engl. *multiprocessor systems*) jesu sistemi s više procesora koji su čvrsto povezani zatvorenim komunikacionim linijama, a nalaze se u istom kućištu. U literaturi se pominju pod imenom paralelni sistemi. Za ovu klasu sistema karakteristično je to da procesori dele istu memoriju i sistemski časovnik. Komunikacija i sinhronizacija između procesora obično se odvijaju preko deljive memorije. Prednosti višeprocessorskih sistema su povećanje brzine, ekonomičnosti i pouzdanosti. Što se tiče povećanja brzine,  $N$  procesora bi u idealnom slučaju radilo  $N$  puta brže, ali je zbog međuprocessorske sinhronizacije to povećanje manje od  $N$ . Ekonomičnost se postiže zato što se dele isti hardverski resursi, memorija, časovnik, DMA kanali, prekidni kontroleri, i jedan izvor napajanja. Povećanje pouzdanosti se postiže tako što će sistem nastaviti da radi i ukoliko otkáže jedan ili više procesora.

Postoje dva koncepta za realizaciju višeprocessorskih sistema: simetrično multiprocesiranje (engl. *symmetric multiprocessing, SMP*), prikazano na slici 1.2, i asimetrično multiprocesiranje (engl. *asymmetric multiprocessing*). U slučaju simetričnog multiprocesiranja, svaki procesor izvršava istu kopiju operativnog sistema, pri čemu te kopije komuniciraju jedna s drugom kad god je potrebno. Svi procesori su ravnopravni, to jest nema odnosa nadređen/podređen (engl. *master/slave*). U idealnom slučaju, svakom procesoru se dodeljuje jedan proces i oni se izvršavaju nezavisno, bez slabljenja performansi. Većina modernih operativnih sistema podržava SMP. U slučaju asimetričnog multiprocesiranja,

svakom procesoru je dodeljen specifičan posao. Postoji jedan glavni procesor (engl. *master*) koji potpuno kontroliše ceo sistem i koji dodeljuje poslove ostalim procesorima (engl. *slaves*). Ova koncepcija se uglavnom koristi u ekstremno brzim i velikim sistemima.



Slika 1.2 Simetrično multiprocesiranje (SMP).

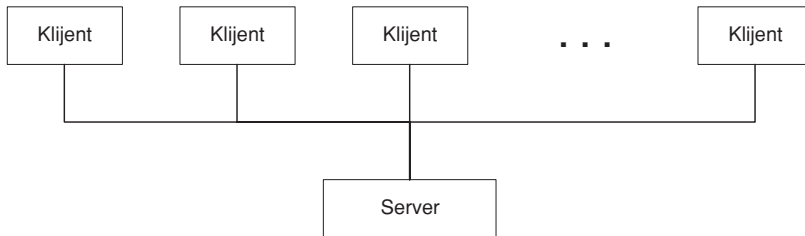
#### [5] Mrežni operativni sistemi

Mrežni operativni sistemi (engl. *network OS*) obezbeđuju okruženje u kome korisnici sa svojih lokalnih mašina mogu pristupiti resursima udaljenih mašina na dva načina: procedurom daljinskog prijavljivanja na sistem (engl. *remote login*) ili razmenom datoteka sa udaljenim sistemom (engl. *remote file transfer*). Posebnu, mnogo kvalitetniju klasu mrežnih operativnih sistema predstavljaju distribuirani sistemi, kod kojih se gubi koncept lokalne i udaljene mašine.

#### [6] Distribuirani sistemi

Distribuirani sistemi (engl. *distributed systems*) predstavljaju kolekciju procesora, tj. računara, koji ne dele zajedničku memoriju i sistemski časovnik. Umesto toga, svaki procesor, tj. računar, ima sopstvenu lokalnu memoriju, a međusobna komunikacija se ostvaruje putem mreže realizovane kao LAN (lokalna mreža) ili WAN (mreža šireg područja). Osim podataka, datoteka i štampača, distribuiraju se i procesi. Četiri glavne prednosti distribuiranih sistema jesu deljenje resursa, ubrzavanje izračunavanja, pouzdanost i komunikacije. Neophodno je naglasiti da distribuirane sisteme karakterišu tri vrste migracija: migracija podataka, migracija izračunavanja i migracija procesa.

Distribuirani sistemi zahtevaju mrežnu infrastrukturu i mogu biti realizovani ili kao klijent/server arhitektura prikazana na slici 1.3, ili kao ravnopravni računarski sistemi koji dele resurse na mreži (engl. *peer-to-peer systems*). U klijent/server arhitekturi, postoje računari koji predstavljaju servere i računari koji koriste njihove usluge – klijenti (engl. *clients*). Postoje dve vrste servera: serveri za izračunavanje (engl. *computing servers*), kojima klijenti šalju zahteve na obradu, i serveri datoteka (engl. *file servers*), koji služe za smeštanje datoteka. Treba svakako istaći i treću vrstu servera, a to su serveri za štampu (engl. *print servers*) koji omogućavaju da klijenti obave štampanje.



Slika 1.3 Klijent/server arhitektura.

## [7] Udruženi sistemi

Udruženi sistemi (engl. *clustered systems*) sastoje se od udruženih računara, to jest od dva ili više nezavisnih računara koji dele diskove i čvrsto su povezani LAN mrežom. Svaki računar u udruženom sistemu naziva se čvor (engl. *node*). Na svakom čvoru izvršava se jedan nivo softvera za udruživanje. Postoje dve vrste udruživanja: asimetrično i simetrično. U asimetričnom udruživanju (engl. *asymmetric clustering*), jedan server (čvor) izvršava aplikaciju, dok ostali – prateći serveri (engl. *monitoring servers*) – prate rad glavnog servera u budnom ali neaktivnom stanju. U slučaju otkaza glavnog servera, jedan od pratećih servera preuće njegovu ulogu. U simetričnom udruživanju (engl. *symmetric clustering*), svi serveri su aktivni i izvršavaju aplikaciju – time se drastično poboljšavaju performanse, ali sistem mora da izdrži otkaz jednog ili više servera kao i u prethodnom slučaju.

## [8] Operativni sistemi za upravljanje u realnom vremenu

Sistemi za upravljanje u realnom vremenu (engl. *real-time systems*) praktično spadaju u klasu specijalnih operativnih sistema. Glavna karakteristika ovih sistema je davanje odziva u propisanom vremenskom intervalu i precizno definisana vremenska ograničenja u kojima moraju da se dogode odzivi. Po pravilu, glavni izvori podataka su senzorski uređaji, te razni prekidači, A/D i D/A konvertori. To su sistemi koji kontrolišu aplikacije specijalne namene, kao što su: specijalni grafički sistemi, sistemi za medicinsku grafiku, sistemi za industrijsku kontrolu, kontrolisanje naučnih eksperimenata itd. Dele se na dve klase: čvrsti sistemi za rad u realnom vremenu (engl. *hard real-time systems*) i meki sistemi za rad u realnom vremenu (engl. *soft real-time systems*). Čvrsti realni sistemi garantuju da će se svi važni poslovi obaviti na vreme. Po pravilu, nemaju sekundarnu memoriju, to jest diskove, jer najčešće rade u teškim uslovima, a kompletan kôd i operativnog sistema i aplikacija čuva se u ROM memoriji. Sam operativni sistem je ograničen, tako da mnoge funkcije operativnih sistema opšte namene uopšte ne postoje. Meki realni sistemi su manje zahtevni, oni obezbeđuju dve klase procesa, a to su obični procesi i procesi u realnom vremenu koji imaju apsolutni prioritet. Praktično, oni su mešavina sistema za rad u realnom vremenu i operativnih sistema opšte namene. Meki realni sistemi su, praktično, sve vrste UNIX i Linux sistema, Windows 2000/XP/2003.

### [9] Ručni sistemi

Ručni sistemi (engl. *handheld systems*) uključuju PDA uređaje (engl. *Personal Digital Assistants*) i mobilne telefone. Glavne karakteristike ovih sistema su mobilnost (korisnici ih mogu nositi svuda sa sobom) i mogućnost priključivanja na mrežu kao što je Internet. Karakterišu ih veoma slabi hardverski resursi, kao što su mala količina memorije (512 KB – 8 MB), procesori skromnih mogućnosti i mali ekrani. Te skromne komponente daju im skromne performanse i mogućnosti. Neki se mogu bežičnim putem priključiti na mrežu, što ih čini veoma atraktivnim.

## Vrste računarskog okruženja

U nastavku teksta ukratko su opisane tri vrste računarskog okruženja (engl. *computing environments*):

### [1] Tradicionalno (engl. *traditional computing*)

Pod tradicionalnim okruženjem podrazumevamo jednokorisničke računare ili sisteme s deljenjem vremena, gde se korisnici preko svojih terminala povezuju na servere.

### [2] Zasnovano na Webu (engl. *Web-based computing*)

Ovo okruženje je zasnovano na umrežavanju po principima globalne mreže (kao što je Internet) na kojoj postoje Web serveri i klijenti u vidu PC računara i malih mobilnih sistema.

### [3] Ugrađeno okruženje (engl. *embedded computing*)

Ugrađeno okruženje je tipično za tvrde *real-time* sisteme koje odlikuje nepostojanje tastature, monitora i diskova. Aplikacija koja se izvršava u realnom vremenu dobija ulazne informacije preko uređaja kao što su senzori, dok se statusne i izlazne informacije prikazuju na LED diodama ili na nekom malom displeju.

## 1.5 OPŠTI PREGLED STRUKTURE OPERATIVNIH SISTEMA

Upravljanje osnovnim resursima računarskog sistema obezbeđuje više funkcionalnih grupa programa namenjenih za:

- upravljanje procesorom;
- upravljanje memorijom;
- upravljanje ulazom i izlazom;
- upravljanje podacima;
- upravljanje sekundarnom memorijom;
- umrežavanje;
- zaštitu;
- korisnički interfejs.



Ponekad je teško precizno reći (usaglasiti) šta se podrazumeva pod delovima operativnog sistema, tj. da li neki upravljački programi pripadaju ili ne pripadaju operativnom sistemu. Prve četiri grupe su fundamentalne i ukratko su opisane u nastavku teksta.

Procesor je jedan od najvažnijih resursa računarskog sistema, mada u pojedinim situacijama neki drugi resursi mogu biti važniji. Upravljanje procesorom može se podeliti na dva nivoa:

- nivo neposredne dodele procesora nekom programu, tj. predaje kontrole nad procesorom nekom programu;
- nivo razrešavanja prioriteta, tj. odlučivanja koji od mogućih programa ima najveći prioritet da bi postao proces i da bi u nekom sledećem trenutku dobio kontrolu nad procesorom.

Pod upravljanjem memorijom podrazumeva se upravljanje radnom, tj. glavnom memorijom računara, kojoj centralni procesor pristupa direktno, radi uzimanja instrukcija ili podataka. Na ovom nivou obavljaju se sledeće funkcije:

- realizacija određene strategije dodeljivanja memorije (redosled dodeljivanja memorije poslovima, statično ili dinamično dodeljivanje, principi dodeljivanja itd.);
- dodeljivanje memorije (algoritmi izbora početne lokacije segmenta koji se dodeljuje);
- sprovođenje određene strategije oslobađanja memorije (ukrupnjavanje manjih oslobođenih delova memorije, oslobađanje delova memorije, redosled obrade zahteva za dodelu memorije).

Na nivou upravljanja uređajima realizuju se sledeće funkcije:

- obezbeđivanje nezavisnosti uređaja (programi treba da budu nezavisni od tipa uređaja koji se koriste za ulazno-izlazne operacije);
- obezbeđivanje efikasnog rada uređaja (pošto U/I operacije predstavljaju često usko grlo računarskog sistema, poželjno je da se što efikasnije preklapaju i izoluju od procesora);
- obezbeđivanje jedinstvenog koncepta analize rada svih uređaja računarskog sistema;
- realizacija određene strategije dodeljivanja uređaja (na primer, redosled dodeljivanja uređaja poslovima, način dodeljivanja uređaja: namenski, s deobom, kao virtuelni itd.);
- dodeljivanje uređaja (fizičko dodeljivanje uređaja, kontrolnih jedinica i kanala poslovima);
- realizacija određene strategije oslobađanja uređaja (na primer, uređaj se oslobađa tek kad se posao kojem je dodeljen završi).

Nivo upravljanja podacima treba da obezbedi softverska sredstva za organizovanje podataka i pristupanje podacima na način koji odgovara korisniku računarskog sistema. Konceptija upravljanja zavisi, pre svega, od vrste podataka i režima njihovog korišćenja. Na ovom nivou realizuju se sledeće funkcije:

- formiranje i brisanje osnovnih struktura podataka (datoteka);
- čitanje datoteka i upisivanje u datoteke;
- obezbeđivanje uslova za simboličko obraćanje datotekama (prema njihovim imenima);
- zaštita podataka od namernog i nenamernog uništavanja (usled otkaza sistema);
- zaštita podataka od neovlašćenog pristupa i korišćenja;
- deoba datoteka (podataka) između više poslova (korisnika).

U literaturi (Lister, Kvaternik), sloj upravljanja podacima često se objedinjuje sa slojem upravljanja sekundarnom memorijom. U tim slučajevima, ovaj sloj obavlja i funkcije organizacije memorijskog prostora na memorijama velikog kapaciteta (najčešće diskovima).

U nastavku teksta opisane su dve koncepcije projektovanja operativnih sistema, a to su monolitna organizacija (engl. *monolithic systems*) i slojevita organizacija (engl. *layered systems*). Takođe uvodimo i treću, veoma savremenu koncepciju, a to je arhitektura mikrojezgra (engl. *microkernel*).

## Monolitni sistemi

U prošlosti, monolitni sistemi su predstavljali najčešću organizaciju operativnih sistema. Ovaj način organizacije operativnih sistema dobio je naziv velika zbrka (engl. *big mess*). Monolitni operativni sistem je realizovan kao skup procedura koje se po potrebi mogu međusobno pozivati, bez ikakvih ograničenja. Korisnički programi upotrebljavaju servise operativnog sistema na sledeći način: parametri sistemskog poziva se smeštaju na određena mesta, kao što su registri procesora ili stek, nakon čega sledi pozivanje jezgra operativnog sistema (engl. *kernel call*). Ova operacija prebacuje procesor iz korisničkog režima rada u sistemski režim rada i kontrolu predaje operativnom sistemu. U sistemskom režimu rada dostupne su neke komande procesora kojima se ne može pristupiti iz korisničkog režima. Posle pozivanja jezgra, operativni sistem preuzima kontrolu i na osnovu parametara poziva određuje koju sistemsku proceduru treba pozvati. Nakon izvršenja procedure, kontrola se vraća korisničkom programu. Operativni sistem ima sledeću strukturu, sastavljenu od tri osnovna skupa programa:

- glavni program koji obrađuje sistemske pozive;
- skup sistemskih procedura koje se pozivaju prilikom sistemskih poziva;
- skup pomoćnih procedura koje koriste sistemske procedure.

## Slojevita (hijerarhijska) realizacija

U slojevitoj realizaciji (engl. *layered system*), operativni sistem se deli na različite slojeve. Slojevi su organizovani hijerarhijski: svaki sloj može da poziva samo funkcije nižih slojeva.

Prvi slojeviti operativni sistem, THE (engl. *Technische Hogeschool Eindhoven*, E.W. Dijkstra), sastojao se od šest slojeva. Slojevi od 0 do 3 predstavljaju jezgro operativnog sistema i rade u sistemskom režimu:

- nulti sloj upravlja procesorom, tj. dodeljuje procesor različitim procesima;
- prvi sloj upravlja memorijom, tj. dodeljuje potrebnu memoriju procesima;
- drugi sloj upravlja komunikacijom između različitih procesa i komandnog interpretera;
- treći sloj obavlja ulazno-izlazne operacije.

Na četvrtom sloju rade korisnički programi, koji se ne brinu o dodeli procesora, dodeljivanju memorije, komandnom interpreteru, ulazno-izlaznim operacijama – sve te operacije obavljaju niži slojevi.

Znači:

- operativni sistem monolitne strukture sastoji se od skupa procedura bez ikakvog grupisanja ili hijerarhije;
- operativni sistem slojevite strukture deli se na više slojeva od kojih svaki ima tačno određenu funkciju (upravlja tačno određenim resursima) i oslanja se isključivo na funkcije nižih slojeva, kojima pristupa pomoću poziva sličnih sistemskim pozivima.

Generalno, slojeviti operativni sistem deli se na određeni broj slojeva, od kojih se svaki gradi na vrhu prethodnog sloja. Najniži sloj (*layer 0*) predstavlja hardver, dok je najviši (*layer N*) – korisnički interfejs. Sa ovakvim modularnim konceptom, slojevi koriste isključivo funkcije i usluge nižih slojeva.

Slojeviti operativni sistem dozvoljava pozivanje (korišćenje) operacija stvaranja i uništavanja procesa samo iz sloja koji se u hijerarhiji nalazi iznad svih slojeva operativnog sistema. To znači da je postojanje procesa isključivo vezano za korisnički sloj. Iako su svi procesi locirani u korisničkom sloju, oni su međusobno jasno razdvojeni zahvaljujući činjenici da svaki od procesa poseduje zaseban adresni prostor – korisnički prostor (engl. *user space*). Na istom principu se zasniva i razdvajanje procesa i operativnog sistema. Ali, razdvajanje korisničkih prostora od adresnog prostora operativnog sistema, tj. sistemskog prostora (engl. *kernel space*), sprečava da se pozivi operacija operativnog sistema zasnivaju na korišćenju poziva potprograma. Zato je neophodno uvođenje posebnog mehanizma sistemskih poziva koji omogućuje prelazak iz korisničkog prostora u sistemski prostor radi pozivanja operacija operativnog sistema. Sistemski pozivi zahtevaju korišćenje specifičnih asemblerskih naredaba i zbog toga se sakrivaju unutar sistemskih potprograma. Neki od sistemskih potprograma, pored sistemskih poziva, sadrže i specifične obrade podataka, kao što je, na

primer, pretvaranje brojeva iz znakovnog u binarni oblik i obrnuto kod formatiranog ulaza ili izlaza. Svaki sistemski potprogram je namenjen za pozivanje jedne od operacija operativnog sistema, namenjenih korisničkom sloju. Ovakve operacije se nazivaju sistemske, kako bi se razlikovale od ostalih (internih) operacija operativnog sistema, namenjenih samo za korišćenje unutar operativnog sistema. Sistemski potprogrami obrazuju sistemsku biblioteku. Prema tome, pozivanje sistemskih operacija svodi se na pozivanje potprograma iz sistemske biblioteke. Sistemsku biblioteku se isporučuje uz operativni sistem kako bi se koristila u postupku povezivanja. U toku povezivanja, bibliotečki potprogrami se vezuju za objektni oblik korisničkog programa radi stvaranja izvršnog oblika korisničkog programa, koji tako postaje spreman za saradnju sa operativnim sistemom.

Zahvaljujući sistemskim potprogramima, tj. sistemskoj biblioteci, operativni sistem predstavlja deo korisničkog programa, iako nije za njega direktno vezan. Zato se može smatrati da obavljanje sistemskih operacija predstavlja sastavni deo izvršavanja korisničkog programa, tj. pripada aktivnosti procesa, vezanog za ovo izvršavanje. To, uz istovremeno postojanje više procesa i nepredvidivost priključivanja, uzrokuje mogućnost da istovremeno postoji više procesa, koji su započeli, a nisu završili svoju aktivnost u okviru operativnog sistema.

## Sistemski pozivi

Aplikacioni programi komuniciraju sa operativnim sistemom pomoću sistemskih poziva (engl. *system calls*), tj. preko operacija (funkcija) koje definiše operativni sistem. Sistemski pozivi se realizuju pomoću sistema prekida: korisnički program postavlja parametre sistemskog poziva na određene memorijske lokacije ili registre procesora, inicira prekid, operativni sistem preuzima kontrolu, uzima parametre, izvrši tražene radnje, rezultat stavlja na određene memorijske lokacije ili u registre i vraća kontrolu programu.

Sistemske pozive često podržava i hardver, tj. procesor, na taj način što razlikuje dva režima rada: korisnički režim (engl. *user mode*) i sistemski režim (engl. *supervisor mode*). Korisnički programi mogu raditi isključivo u korisničkom režimu rada procesora, sistemski režim rada je predviđen za operativni sistem. Ako korisnički program pokuša da izvrši neku operaciju koja je dozvoljena samo u sistemskom režimu rada, kontrola se predaje operativnom sistemu. Prilikom sistemskih poziva procesor prelazi iz korisničkog režima rada u sistemski, a vraća se u korisnički režim posle obrade poziva.

Sistemski pozivi obezbeđuju interfejs između programa koji se izvršava i operativnog sistema. Generalno, realizuju se na asemblerskom jeziku, ali noviji viši programski jezici, poput jezika C i C++, takođe omogućavaju realizaciju sistemskog poziva. Program koji se izvršava može proslediti parametre operativnom sistemu na tri načina:

- prosleđivanjem parametara u registrima procesora;
- postavljanjem parametara u memorijskoj tabeli, pri čemu se adresa tabele prosleđuje u registru procesora;
- postavljanjem parametara na vrh steka (push), koje operativni sistem „skida“ (pop).

## Arhitektura mikrojezgra

Mikrojezgro (engl. *microkernel*) predstavlja veoma savremen koncept u realizaciji savremenih operativnih sistema. Osnovna zamisao je napraviti minimalno i pouzdano jezgro visokih performansi, a sve ostale funkcije jezgra potisnuti u takozvani korisnički prostor (engl. *user space*). Korisnički moduli međusobno komuniciraju slanjem poruka (engl. *message passing*).

U dobre osobine arhitekture mikrojezgra spadaju:

- jednostavno proširivanje i optimizacija jezgra;
- jednostavno dodavanje novih modula bez uticaja na osnovno jezgro;
- jednostavna prenosivost na drugu računarsku arhitekturu;
- veća pouzdanost (manje koda se izvršava u režimu jezgra);
- veća sigurnost.

## Virtuelne mašine

Iako smo operativni sistem definisali kao virtuelnu mašinu, IBM je razvio posebnu softversku strukturu koja ima velike prednosti pri realizaciji novih operativnih sistema i koja se takođe naziva virtuelna mašina.

Strukturu virtuelnih mašina IBM definiše na sledeći način: na najnižem nivou se nalazi hardver, a iznad hardvera monitor virtuelnih mašina (engl. *virtual machine monitor*), to jest poseban sistem koji obezbeđuje niz virtuelnih mašina (tačnih kopija hardvera). Zatim se na te virtuelne mašine mogu instalirati različiti operativni sistemi. Odgovarajući operativni sistemi primaju systemske pozive korisničkih programa, a hardverske operacije koje ti operativni sistemi šalju prema svojim virtuelnim mašinama prihvata monitor virtuelnih mašina i realizuje ih u skladu s hardverom ispod sebe.

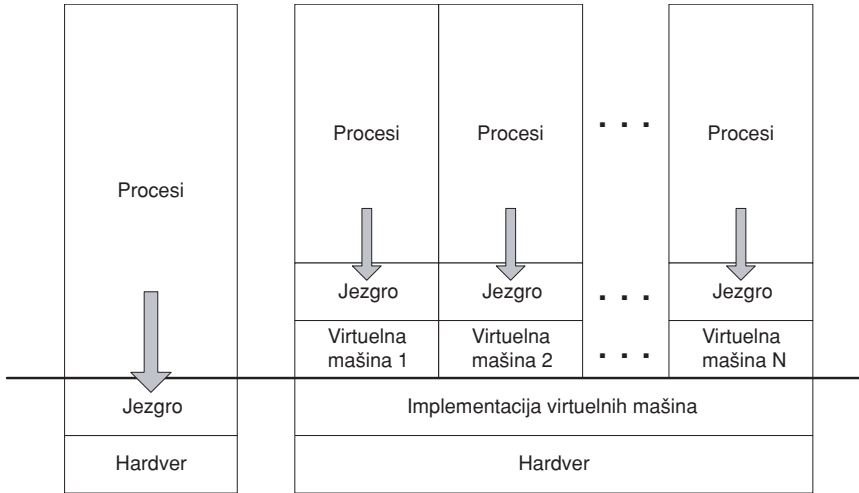
Virtuelna mašina je zasnovana na slojevitoj organizaciji i tretira realni hardver i realno jezgro kao da su hardver za operativni sistem koji predstavljaju. Virtuelna mašina obezbeđuje identičan interfejs kao da je realni hardver ispod virtuelne mašine, a ne čitav niz slojeva softvera.

Operativni sistem stvara iluziju o višestrukim procesima koji se izvršavaju na svom virtuelnom procesoru i svojoj virtuelnoj memoriji. Virtuelne mašine obezbeđuju kompletnu zaštitu sistemskih resursa, pošto su sve virtuelne mašine međusobno izolovane, i ne mogu direktno deliti resurse.

Koncept virtuelne mašine prikazan je na slici 1.4.

Virtuelne mašine su perfektan razvojni sistem za realizaciju novih operativnih sistema, jer se razvoj realizuje na virtuelnoj mašini, a ne na realnoj, može lako da se kontroliše, mogu se ispravljati greške i nikako se ne mogu narušiti normalne systemske operacije, jer će sistem koji nudi virtuelne mašine raditi bez obzira na stanje pojedinih virtuelnih mašina.

Sistem virtuelnih mašina ne može se lako realizovati zato što treba egzaktno simulirati hardver koji reprezentuje virtuelna mašina.



Slika 1.4 Virtuelne mašine.

## 1.6 OPERATIVNI SISTEMI UNIX I LINUX

UNIX je stabilan, moćan i fleksibilan operativni sistem visokih performansi, pogodan za izvršavanje važnih aplikacija. UNIX je čvrsto povezan s mrežnim servisima TCP/IP protokola, čime je u potpunosti promjenjena slika UNIX servera i radnog okruženja iz prošlosti. Umesto servera s klasičnim serijskim terminalima, UNIX server se nalazi u mreži, pri čemu s radnim stanicama ostvaruje vezu preko LAN/WAN mreže i skupa protokola TCP/IP. Većina velikih svjetskih proizvođača računara razvija specifičnu varijantu operativnog sistema UNIX, što ukazuje na njegov kvalitet, popularnost i rasprostranjenost. Većina UNIX sistema, poput IBM AIX i Sun Solaris, komercijalna je – korisnik mora da plati licencu za upotrebljavanje, a izvorni kôd nije raspoloživ. To su razlozi narastajuće popularnosti Linux sistema koji zadržava većinu dobrih osobina sistema UNIX, a dodatno se odlikuje raspoloživim izvornim kodom i praktično besplatnim korišćenjem. Zbog toga danas većina proizvođača računara – osim sopstvene komercijalne verzije UNIX sistema – nudi i podršku za Linux. Linux se najčešće koristi kao operativni sistem na radnim stanicama ili serverima u manjoj ili srednjoj klasi servera, a jedna od oblasti dominantne primene, u kojoj veliki broj korisnika podržava i promovira Linux kao bazični server, jesu Internet usluge.

### Istorijat operativnog sistema UNIX

Razvoj operativnog sistema UNIX počeo je sredinom šezdesetih godina u AT&T Bell laboratorijama, u saradnji s kompanijom General Electric i tehnološkim institutom Massachusetts. Projekat, to jest operativni sistem MULTICS (*Multiplexed Information and Computing Service*), predstavljao je interaktivni operativni sistem namenjen

da opslužuje veliki broj korisnika čiji su terminali direktnim serijskim ili modemskim komunikacionim kanalima povezani na centralizovani server. Takav koncept operativnog sistema bio je preambiciozan za tadašnji stepen razvijenosti hardvera, pri čemu se prvenstveno misli na procesorsku snagu i količinu systemske memorije. MULTICS nije doživeo praktičnu primenu jer se posle nekoliko godina razvoja pokazao kao preskup i zahtevan projekat od koga su AT&T Bell laboratorije odustale. Bez obzira na to, teorijska i praktična rešenja projekta našla su primenu u mnogim operativnim sistemima. Konkretno, MULTICS je preteča UNIX sistema koji se smatra jednim od najkvalitetnijih i najrasprostranjenijih operativnih sistema, na čijem se razvoju radi preko 30 godina, s tendencijom dalje egzistencije i usavršavanja.

Dva fundamentalna imena vezana za razvoj operativnog sistema UNIX svakako su Ken Tompson (Ken Thompson), programer sistema MULTICS u Bell laboratorijama, i Denis Riči (Dennis Ritchie), poznatiji kao tvorac programskog jezika C. Godine 1969, Ken Tompson je započeo razvoj novog operativnog sistema za DEC PDP-7 računar, napravivši redukovani MULTICS, tj. UNICS (*Uniplexed Information and Computing Service*). Radi lakšeg izgovora i pisanja, ime UNICS je kasnije evoluiralo u UNIX. UNIX je prvobitno napisan na asemblerskom jeziku, a samim tim bio je potpuno zavisan od klase procesora za koji se realizuje. Godine 1971, Riči je napisao programski jezik C kao viši programski jezik koji omogućava systemsko programiranje, a zatim je s Tompsonom preveo kôd UNIX sistema na C, što se može smatrati prekretnicom i jednim od najznačajnijih poteza u razvoju operativnih sistema. Zahvaljujući jeziku C, UNIX je mogao biti prenesen na razne računarske arhitekture s vrlo malo programskih modifikacija, što je svakako bio ključ uspeha i popularnosti operativnog sistema UNIX.

Nakon prevoda na jezik C, autori su u cilju daljeg unapređenja prosledili izvorni kôd UNIX sistema univerzitetima širom Amerike, pri čemu su programeri sa univerziteta Berkeley u Kaliforniji dominantno uticali na dalji razvoj. Nakon brojnih modifikacija koda nastao je BSD UNIX (*Berkeley Software Distribution*), koji je takođe u izvornom obliku distribuiran američkim univerzitetima radi daljeg usavršavanja. Najznačajniji doprinos grupe koja je realizovala BSD UNIX odnosi se na softver za umrežavanje, koji omogućava da operativni sistem funkcioniše u LAN i WAN mrežama. BSD UNIX mrežna rešenja prihvatili su brojni proizvođači računara koji su razvijali sopstvene varijante operativnog sistema UNIX:

- SunOS, kompanije Sun Microsystems, zasnovan na sistemu BSD UNIX v4.2;
- System V UNIX, kompanije AT&T;
- XENIX, kompanije Microsoft, razvijen za PC računare sa Intel procesorima.

Godine 1988, kombinovanjem dobrih osobina Sun OS/BSD, AT&T System V Release 3 i XENIX operativnih sistema nastala je nova varijanta UNIX sistema – System V Release 4 (SVR4), koja je ubrzo postala de facto standard i osnova za dalji razvoj UNIX sistema.

Sledeći korak u razvoju predstavlja pokušaj standardizacije i realizacije međusobne kompatibilnosti raznih vrsta UNIX sistema.

## Vrste UNIX sistema

Danas postoji veliki broj vrsta UNIX sistema koje su zasnovane ili na industrijskom standardu SVR4 ili na BSD distribuciji. One vrste UNIX sistema koje potiču od istog standarda međusobno su veoma slične, tako da se na osnovu nekih indikatora (najčešće na osnovu sistemskih komandi) može odrediti koji je standard korišćen kao osnova za izgradnju i razvoj specifičnog UNIX sistema. Na sistemima čija je osnova System V, komanda za štampanje je `lp`, na sistemima zasnovanim na distribuciji BSD – `lpr`. Komanda koja opisuje tekuće procese na System V sistemima jeste `ps -ef`, dok BSD koristi `ps -aux`. Razlike postoje i u drugim komandama.

UNIX je sada zaštićeno ime u vlasništvu organizacije Open Group ([www.opengroup.org](http://www.opengroup.org)), koja je definisala standard POSIX (*Portable Operating System Interface*) u cilju povećanja kompatibilnosti raznih vrsta UNIX sistema. Svaki proizvođač svoju UNIX distribuciju naziva jedinstvenim imenom: Solaris (Sun Microsystems), AT&T UNIX, AIX (IBM), HP-UX (Hewlett-Packard), Tru64UNIX (Compaq). Iako je većina UNIX sistema osmišljena da radi uglavnom na određenoj arhitekturi sa određenom klasom procesora (najčešće su to procesori matične kompanije), postoje varijante koje rade na više klasa procesora. Na primer, Solaris – koji je prvenstveno namenjen Sun računarima – postoji i u verziji za PC arhitekturu, tj. za Intel procesore, a Linux, pored PC arhitekture podržava i druge arhitekture kao što su Sun, Apple Macintosh i IBM. Kada se savlada rad na jednoj vrsti UNIX sistema, prelazak na neku drugu verziju relativno je lak.



## Linux

Jedna od poslednjih varijanti operativnih sistema UNIX, čiji je razvoj započeo Linus Torvalds 1991. godine, na Univerzitetu u Helsinkiju, jeste Linux. Torvalds je svoj operativni sistem – koji objedinjuje oba standarda, SVR4 i BSD – objavio na Internetu i podsticao druge programere širom sveta da se priključe njegovom daljem razvoju. Ubrzo je Linux postao veoma popularan među računarskim entuzijastima koji su tražili alternativno rešenje za postojeće operativne sisteme za PC računare (DOS, Windows). Linux je zbog svoje koncepcije stabilnog a jeftinog operativnog sistema doživeo veliku ekspanziju i popularnost. Simbol Linux sistema je mali pingvin (Tux).

Linux je prvobitno bio namenjen 32-bitnim Intel x86 mikroprocesorima (počevši od 80386), na kojima može funkcionisati kao radna stanica (engl. *workstation*) ili kao server. Jezgro Linux sistema kasnije je modifikovano i prilagođeno procesorima koji ne pripadaju Intel x86 klasi, među kojima treba istaći Intel IA-64, DEC Alpha, SUN SPARC/UltraSPARC, Motorola 68000, MIPS, PowerPC i IBM mainframe S/390. Može se konstatovati da današnji Linux podržava širi spektar procesora i računarskih arhitektura od svih drugih operativnih sistema.





## GNU/Linux i Open Source

Veliki deo komponenata operativnom sistemu Linux dodali su nezavisni programeri i programeri GNU projekta ([www.gnu.org](http://www.gnu.org)), koji pripada fondaciji za besplatan softver (*Free Software Foundation*, FSF). Svi operativni sistemi GNU/Linux koriste Linux jezgro kao fundamentalni deo koji kontroliše interakciju između hardvera i aplikacija, i GNU aplikacije kao dodatne komponente operativnog sistema.

Linux je raspoloživ kao besplatan operativni sistem pod licencom GNU GPL (*GNU General Public License*), što važi i za neke druge vrste UNIX sistema, kao što su FreeBSD i NetBSD. Linux je softver sa otvorenim izvornim kodom (engl. *Open Source*), što znači da je izvorni kôd javno raspoloživ i da se može modifikovati tako da odgovara specifičnim potrebama. Linux se može slobodno distribuirati među korisnicima. Ovakav koncept je potpuno suprotan konceptu komercijalnog softvera, gde izvorni kôd nije dostupan i svaki korisnik mora da plati licencu za korišćenje. Komercijalni softver je zasnovan na zakonima o autorskim pravima (engl. *copyright laws*); njima se propisuju ograničenja koja korisnici softvera imaju u odnosu na izvorni kôd, korišćenje i dalje distribuiranje softvera. Linux se besplatno može preuzeti s različitih Web lokacija.

## Distribucije Linuxa

Brojne profitne i neprofitne organizacije čine Linux raspoloživim u obliku distribucija, tj. različitih kombinacija jezgra, sistemskog softvera i korisničkih aplikacija. Većina distribucija sadrži kolekciju CD/DVD medijuma na kojima se nalaze operativni sistem, izvorni kôd, detaljna dokumentacija, i štampana uputstva za instaliranje i upotrebu sistema. Cene ovakvih distribucija su u većini slučajeva simbolične, osim ako se u distribuciji nalazi komercijalan softver ili je distribucija specifične namene.

Osnovna komponenta svake distribucije Linuxa jeste jezgro operativnog sistema. Osim jezgra i sistemskog softvera, u distribuciji se nalaze i instalacioni alati, softver za podizanje operativnog sistema (engl. *boot loader*), razne korisničke aplikacije (kancelarijski paketi, softver za obradu bitmapiranih slika) i serverski paketi. Većina distribucija ima grafičko korisničko okruženje – poput Windows sistema – dok su neke distribucije namenjene administratorima sistema i programerima, bliske tradicionalnom UNIX okruženju.

U poznatije distribucije Linuxa spadaju: Debian GNU/Linux (<http://www.debian.org>), Linux Mandrake (<http://linux-mandrake.com/en>), Red Hat Linux (<http://www.redhat.com>), Slackware Linux (<http://www.slackware.com>) i SuSE Linux (<http://www.suse.com>).

## Opšti pregled strukture Linux sistema

Linux je višekorisnički, višeprocensni operativni sistem s potpunim skupom UNIX kompatibilnih alata, projektovan tako da poštuje relevantne POSIX standarde. Linux sistemi podržavaju tradicionalnu UNIX semantiku i potpuno implementiraju standardni UNIX mrežni model.

Operativni sistem Linux sastoji se od jezgra, sistemskog softvera, korisničkih aplikacija, programskih prevodilaca i njihovih odgovarajućih biblioteka (GCC – GNU C Compiler i C biblioteka za Linux) i dokumentacije. Sadržaj konkretne distribucije Linuxa definisan je sadržajem instalacionih medijuma, koji u slučaju nekih Linux sistema uključuju razne FTP lokacije širom sveta.

Jezgro operativnog sistema omogućava konkurentno (paralelno) izvršavanje procesa, dodeljuje im memoriju i druge resurse i obezbeđuje mehanizam za ostvarivanje usluga operativnog sistema. Jezgro štiti korisničke procese od direktnog pristupa hardveru – procesi pristupaju hardveru korišćenjem sistemskih poziva jezgra, čime se obezbeđuje zaštita između samih korisnika. Sistemski programi koriste jezgro u cilju implementacije različitih usluga operativnog sistema.

Svi programi, uključujući i sistemske, funkcionišu na nivou iznad jezgra, što se naziva korisnički režim rada, dok se sistemske aktivnosti – poput pristupa hardveru obavljaju na nivou jezgra, tj. u sistemskom režimu rada (engl. *supervisory mode*). Sistemski i aplikativni programi razlikuju se po nameni: aplikacije su namenjene za razne korisne aktivnosti (kao što su obrada teksta i slika), dok su sistemski programi namenjeni za rad sa sistemom i administraciju. Na primer, program za obradu teksta je korisnička aplikacija, dok je komanda `mount` sistemski program. Razlike između korisničkih i sistemskih programa ponekad su veoma male, i značajne su samo u strogim kategorizacijama softvera.

## 1.7 PITANJA I ZADACI

- 1.1 Koje osnovne ciljeve operativni sistem mora da postigne pri posredovanju između korisnika i računarskog sistema?
- 1.2 Navedite osnovne funkcije operativnog sistema.
- 1.3 Navedite osnovne karakteristike operativnih sistema.
- 1.4 a. Definišite efikasnost i premašenje operativnog sistema.  
b. Na osnovu kojih kriterijuma se određuje efikasnost operativnog sistema?
- 1.5 Navedite razloge zbog kojih se operativni sistemi najčešće pišu na programskom jeziku C.
- 1.6 Jedna od poželjnih osobina operativnog sistema je visoka efikasnost. U kom slučaju se može dozvoliti da operativni sistem „pregazi“ ovaj princip i nepotrebno „pojede“ resurse?
- 1.7 Zašto ne postoje višekorisnički jednoproceni operativni sistemi?
- 1.8 Navedite primere (a) jednokorisničkih jednoprocenih, (b) jednokorisničkih višeprocenih i (c) višekorisničkih višeprocenih operativnih sistema.
- 1.9 Navedite najznačajnije karakteristike (a) operativnih sistema za paketnu obradu, (b) interaktivnih operativnih sistema, (c) operativnih sistema s deljenjem vremena, (d) operativnih sistema za rad u realnom vremenu i (e) distribuiranih operativnih sistema.
- 1.10 Koja je osnovna razlika između upitnih i transakcionih sistema?

- 1.11** U kom slučaju se korišćenje sistema s deljenjem vremena smatra boljim rešenjem od korišćenja personalnog računara, to jest radne stanice za jednog korisnika?
- 1.12** Koja je razlika između simetričnog (SMP) i asimetričnog multiprocesiranja u više-procesorskim sistemima?
- 1.13** Koji je osnovni problem karakterističan za projektovanje *real-time* operativnih sistema?
- 1.14** Bazu podataka s pratećom aplikacijom treba postaviti na četiri servera. Kako se mogu udružiti ovi serveri ukoliko je potrebno obezbediti (a) visoku pouzdanost, (b) visoke performanse?
- 1.15** Koje funkcionalne grupe programa čine operativni sistem?
- 1.16** Zašto se u teorijskom modelu operativnog sistema (engl. *paper-model OS*) sloj upravljanja datotekama nalazi iznad slojeva upravljanja ulazom i izlazom, i upravljanja memorijom?
- 1.17** Koja je razlika između monolitnih i hijerarhijskih operativnih sistema?
- 1.18** a. Koja je osnovna karakteristika operativnih sistema s arhitekturom mikrojezgra?  
b. Zašto su ovakvi sistemi pouzdaniji od monolitnih?
- 1.19** Objasnite kako razdvajanje korisničkog i sistemskog režima funkcioniše kao poseban oblik zaštite sistema.
- 1.20** Koja od sledećih instrukcija ne mora biti privilegovana: (a) postavljanje vrednosti tajmera, (b) očitavanje sistemskog časovnika, (c) brisanje memorije, (d) isključivanje prekida (e) prelazak iz korisničkog režima u sistemski?
- 1.21** Sledeće instrukcije su privilegovane: (a) prelazak u korisnički režim, (b) prelazak u sistemski režim, (c) čitanje iz zaštićene memorije, (d) upis u zaštićenu memoriju, (e) prihvatanje instrukcije iz zaštićene memorije, (f) sprečavanje prekida koji izaziva tajmer, (g) omogućavanje prekida koji izaziva tajmer. Odaberite minimalni skup instrukcija koje moraju biti privilegovane, tako da se ne naruši neophodan stepen zaštite sistema.
- 1.22** Koja je uloga sistemskih poziva?
- 1.23** Koja je uloga sistemskih programa?
- 1.24** Kako virtuelna mašina odgovara na sistemske pozive?
- 1.25** Kojoj procesorskoj arhitekturi je namenjen operativni sistem Linux?
- 1.26** a. U kom obliku je raspoloživ Linux?  
b. Da li je izvorni kôd operativnog sistema javno dostupan ili nije?  
c. Da li se Linux može slobodno distribuirati?
- 1.27** a. Šta ulazi u sastav jedne distribucije Linuxa?  
b. Navedite neke od značajnijih distribucija Linux sistema.
- 1.28** Kakav je operativni sistem Linux sa aspekta broja korisnika koji istovremeno mogu upotrebljavati sistem i broja procesa koji se mogu istovremeno izvršavati?