Uvodu u informatiku – Sistemski softver

Danijela Simić

algoritmi i izračunljivost

1. oktobar 2024.

Sadržaj

Uvod

Uvod

Sistemskog softver:

- operativni sistem:

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja
- antivirusni program

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja
- antivirusni program
- alata za održavanje, konfiguraciju i optimizaciju sistema

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja
- antivirusni program
- alata za održavanje, konfiguraciju i optimizaciju sistema
- programi za učitavanje operativnog sistema

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja
- antivirusni program
- alata za održavanje, konfiguraciju i optimizaciju sistema
- programi za učitavanje operativnog sistema
- ugrađeni programi poput BIOS-a

Uvod

Sistemskog softver:

- operativni sistem:
  - skup programa koji obezbeđuju efikasnu i bezbednu kontrolu računarskih resursa.
  - obezbeđuje mehanizme korišćenja tih resursa od strane aplikativnih programa
  - omogućava komunikaciju korisnika sa računarom podsredstvom ulazno-izlaznih uređaja
- antivirusni program
- alata za održavanje, konfiguraciju i optimizaciju sistema
- programi za učitavanje operativnog sistema
- ugrađeni programi poput BIOS-a
- različitih firmvera za uređaje

Uvod

Sistemskog softver:

- programerski alati poput previodica, linkera i debagera

Uvod

Sistemskog softver:

- programerski alati poput previodica, linkera i debagera
- mrežne servere

Uvod

Sistemskog softver:

- programerski alati poput previodica, linkera i debagera
- mrežne servere
- editora teksta, jednostavnih igara, multimedijalnih programa?

Operativni sistem

Operativni sistem

- Komponente operativnog sistema se mogu nalaziti u ROM memoriji (u novije vreme EEPROM) ili u spoljašnjim memorijama (danas tipično na hard ili SSD disku).

Operativni sistem

- Komponente operativnog sistema se mogu nalaziti u ROM memoriji (u novije vreme EEPROM) ili u spoljašnjim memorijama (danas tipično na hard ili SSD disku).
- Po uključivanju računara automatski učitavaju u operativnu memoriju (RAM).

Operativni sistem

- Komponente operativnog sistema se mogu nalaziti u ROM memoriji (u novije vreme EEPROM) ili u spoljašnjim memorijama (danas tipično na hard ili SSD disku).
- Po uključivanju računara automatski učitavaju u operativnu memoriju (RAM).
- Započinju sa izvršavanjem na procesoru računara.

Operativni sistem

Ranija vremena — računari su imali fiksirani, fabrički ugrađeni operativni sistem koji se nalazio u ROM memoriji računara.

Operativni sistem

Ranija vremena — računari su imali fiksirani, fabrički ugrađeni operativni sistem koji se nalazio u ROM memoriji računara.

Danas – operativni sistem koji je izmenjiv, jer se nalazi u spoljašnjoj memoriji.

- Korisnik ima mogućnost izbora operativnog sistema koji će koristiti.

Operativni sistem

Ranija vremena — računari su imali fiksirani, fabrički ugrađeni operativni sistem koji se nalazio u ROM memoriji računara.

Danas – operativni sistem koji je izmenjiv, jer se nalazi u spoljašnjoj memoriji.

- Korisnik ima mogućnost izbora operativnog sistema koji će koristiti.
- Korisnik ima mogućnost nadogradnje verzije operativnog sistema i pratećeg softvera.

Operativni sistemi

Ne–slobodna licenca:

- Windows (kompanija Microsotf)
- macOS (UNIX-zasnovan, kompanija Apple)

Operativni sistemi

Ne–slobodna licenca:

- Windows (kompanija Microsotf)
- macOS (UNIX-zasnovan, kompanija Apple)

Slobodna licenca:

- Sistemi zasnovani na operativnom sistemu *UNIX*
- ReactOS (kompatibilni sa Windows)

Operativni sistemi

Ne–slobodna licenca:

- Windows (kompanija Microsotf)
- macOS (UNIX-zasnovan, kompanija Apple)

Slobodna licenca:

- Sistemi zasnovani na operativnom sistemu *UNIX*
- ReactOS (kompatibilni sa Windows)

Pametni telefoni (u širokoj upotrebi su ne-slobodne licence):

- Android (kompanija Google)
- iOS (kompanija Apple)

Operativni sistemi

Ne–slobodna licenca:

- Windows (kompanija Microsotf)
- macOS (UNIX-zasnovan, kompanija Apple)

Slobodna licenca:

- Sistemi zasnovani na operativnom sistemu *UNIX*
- ReactOS (kompatibilni sa Windows)

Pametni telefoni (u širokoj upotrebi su ne-slobodne licence):

- Android (kompanija Google)
- iOS (kompanija Apple)

Operativni sistem

Struktura operativnog sistema

Struktura operativnog sistema

Jezgro

Glavni deo operativnog sistema naziva se *jezgro* .

Struktura operativnog sistema

Jezgro

Glavni deo operativnog sistema naziva se *jezgro* .

Zadužen za upravljanje računarskim resursima, komunikaciju sa hardverom, kao i obezbeđivanje interfejsa ka korisničkim programima.

Struktura operativnog sistema

Jezgro

Glavni deo operativnog sistema naziva se *jezgro* .

Zadužen za upravljanje računarskim resursima, komunikaciju sa hardverom, kao i obezbeđivanje interfejsa ka korisničkim programima.

> Korisnički programi pristupaju resursima računara isključivo putem *interfejsa jezgra*.

Struktura operativnog sistema

Jezgro

Glavni deo operativnog sistema naziva se *jezgro* .

Zadužen za upravljanje računarskim resursima, komunikaciju sa hardverom, kao i obezbeđivanje interfejsa ka korisničkim programima.

> Korisnički programi pristupaju resursima računara isključivo putem *interfejsa jezgra*.

Sistemski poziv

Interfejs jezgra se obično sastoji iz skupa funkcija koje se nazivaju *sistemski pozivi*.

Sistemski poziv

Sistemski poziv obezbeđuje bezbedan transfer kontrole jezgru operativnog sistema, kako bi ono moglo da korisničkom programu pruži traženu uslugu.

Sistemski poziv

Interfejs jezgra se obično sastoji iz skupa funkcija koje se nazivaju *sistemski pozivi*.

Sistemski poziv

Sistemski poziv obezbeđuje bezbedan transfer kontrole jezgru operativnog sistema, kako bi ono moglo da korisničkom programu pruži traženu uslugu.

- Sistemski pozivi su obično implementirani na nivou asemblerskog jezika.

Sistemski poziv

Interfejs jezgra se obično sastoji iz skupa funkcija koje se nazivaju *sistemski pozivi*.

Sistemski poziv

Sistemski poziv obezbeđuje bezbedan transfer kontrole jezgru operativnog sistema, kako bi ono moglo da korisničkom programu pruži traženu uslugu.

- Sistemski pozivi su obično implementirani na nivou asemblerskog jezika.
- Sistemske biblioteke — omogućavaju programerima da upućuju zahteve jezgru operativnog sistema iz različitih programskih jezika.

Korisnički interfejs

Korisnički interfejs je korisnički program (ili skup programa) koji se pokreće odmah nakon što se korisnik prijavi na sistem, a koji omogućava korisniku da izdaje komande, pokreće korisničke programe i prati njihov rad, kao i da koristi podatke na spoljnim memorijama.

Operativni sistem

Programi i procesi

Programi i procesi

Proces

Proces predstavlja program u izvršenju.

Programi i procesi

Proces

Proces predstavlja program u izvršenju.

Program – može postojati negde u memoriji i ne izvršavati se.

Procesi

Proces – kontekst u kome se program izvršava, a koji kreira operativni sistem:

- sadrže informacije o **trenutnom stanju** u kome se program nalazi prilikom izvršenja;

Procesi

Proces – kontekst u kome se program izvršava, a koji kreira operativni sistem:

- sadrže informacije o **trenutnom stanju** u kome se program nalazi prilikom izvršenja;
- informacije o **resursima** koji su programu dodeljeni (poput memorije, otvorenih fajlova, kanala za komunikaciju sa drugim procesima i sl.)

Procesi

Procesi se unutar operativnog sistema obično identifikuju brojevima procesa (engl. *process identifier* (PID)).

Procesi

Procesi se unutar operativnog sistema obično identifikuju brojevima procesa (engl. *process identifier* (PID)).

Pokretanje programa:

- operativni sistem najpre inicijalizuje proces i dodeljuje mu jedinstven PID;

Procesi

Procesi se unutar operativnog sistema obično identifikuju brojevima procesa (engl. *process identifier* (PID)).

Pokretanje programa:

- operativni sistem najpre inicijalizuje proces i dodeljuje mu jedinstven PID;
- dodeljuje memoriju;

Procesi

Procesi se unutar operativnog sistema obično identifikuju brojevima procesa (engl. *process identifier* (PID)).

Pokretanje programa:

- operativni sistem najpre inicijalizuje proces i dodeljuje mu jedinstven PID;
- dodeljuje memoriju;
- u dodeljenu memoriju učitava programski kôd programa i inicijalizuje njegove podatke;

Procesi

Procesi se unutar operativnog sistema obično identifikuju brojevima procesa (engl. *process identifier* (PID)).

Pokretanje programa:

- operativni sistem najpre inicijalizuje proces i dodeljuje mu jedinstven PID;
- dodeljuje memoriju;
- u dodeljenu memoriju učitava programski kôd programa i inicijalizuje njegove podatke;
- dodeljuje procesor (program kreće da se izvršava).

Procesi

Proces može biti kreiran od strane jezgra operativnog sistema ili od strane drugog procesa.

Procesi

Proces može biti kreiran od strane jezgra operativnog sistema ili od strane drugog procesa.

> Proces koji kreira drugi proces ćemo nazivati *roditeljski proces* (engl. *parent process*).

Dodatni resursi:

- ako proces želi da otvori neki fajl na disku: operativni sistem će kreirati strukturu podataka koja će sadržati informacije o lokaciji otvorenog fajla na disku, broju bajta do koga se stiglo sa čitanjem, bafere koji sadrže pročitane bajtove i sl.

Procesi

Proces može biti kreiran od strane jezgra operativnog sistema ili od strane drugog procesa.

> Proces koji kreira drugi proces ćemo nazivati *roditeljski proces* (engl. *parent process*).

Dodatni resursi:

- ako proces želi da otvori neki fajl na disku: operativni sistem će kreirati strukturu podataka koja će sadržati informacije o lokaciji otvorenog fajla na disku, broju bajta do koga se stiglo sa čitanjem, bafere koji sadrže pročitane bajtove i sl.
- Procesu će, po otvaranju fajla, biti prosleđena *ručka* — neka vrsta identifikatora otvorenog fajla.

Proces – završetak

- Jedan način – završi izvršavanje programskog kôda.

Proces – završetak

- Jedan način – završi izvršavanje programskog kôda.
- Vraća kontrolu operativnom sistemu.

Proces – završetak

- Jedan način – završi izvršavanje programskog kôda.
- Vraća kontrolu operativnom sistemu.
- Dealocira resurse dodeljene procesu i uklanja pridružene strukture podataka iz memorije.

Proces – završetak

- Jedan način – završi izvršavanje programskog kôda.
- Vraća kontrolu operativnom sistemu.
- Dealocira resurse dodeljene procesu i uklanja pridružene strukture podataka iz memorije.
- Drugi način je da program bude prekinut protiv svoje volje — operativni sistem (neka nedozvoljena radnja) ili korisnik (šalje signal operativnom sistemu).

Operativni sistem

Pokretanje i zaustavljanje operativnog sistema

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Programski brojač

Registar procesora koji sadrži adresu instrukcije koju sledeću treba izvršiti.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Programski brojač

Registar procesora koji sadrži adresu instrukcije koju sledeću treba izvršiti.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Programski brojač

Registar procesora koji sadrži adresu instrukcije koju sledeću treba izvršiti.

- Programski brojač je fabrički podešen na neku fiksiranu početnu vrednost, pa je potrebno obezbediti da počev od te adrese imamo program koji prvi treba da se izvrši po uključivanju računara.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Programski brojač

Registar procesora koji sadrži adresu instrukcije koju sledeću treba izvršiti.

- Programski brojač je fabrički podešen na neku fiksiranu početnu vrednost, pa je potrebno obezbediti da počev od te adrese imamo program koji prvi treba da se izvrši po uključivanju računara.
  Obično ROM memorije — sadrže taj početni program koji se prvi izvršava.

Pokretanje i zaustavljanje operativnog sistema

- Procesor odmah započinje sa radom.

Programski brojač

Registar procesora koji sadrži adresu instrukcije koju sledeću treba izvršiti.

- Programski brojač je fabrički podešen na neku fiksiranu početnu vrednost, pa je potrebno obezbediti da počev od te adrese imamo program koji prvi treba da se izvrši po uključivanju računara.
  Obično ROM memorije — sadrže taj početni program koji se prvi izvršava.

  Tradicionalno – **BIOS** (engl. *basic-input-output-system*).
  Modernija varijanta – **UEFI** (engl. *unified extensible firmware interface*).

Pokretanje i zaustavljanje operativnog sistema

- BIOS – inicijalizuje hardverske komponente i izvrši neke osnovne testove ispravnosti,

Pokretanje i zaustavljanje operativnog sistema

- BIOS – inicijalizuje hardverske komponente i izvrši neke osnovne testove ispravnosti,
- Kontrolu predaje dalje – punilac (engl. *boot loader*).

Pokretanje i zaustavljanje operativnog sistema

- BIOS – inicijalizuje hardverske komponente i izvrši neke osnovne testove ispravnosti,
- Kontrolu predaje dalje – punilac (engl. *boot loader*).
- Punilac se, kao i sâm operativni sistem, nalazi na nekoj od spoljnih memorija (tipično na hard ili SSD disku).

Pokretanje i zaustavljanje operativnog sistema

- BIOS – inicijalizuje hardverske komponente i izvrši neke osnovne testove ispravnosti,
- Kontrolu predaje dalje – punilac (engl. *boot loader*).
- Punilac se, kao i sâm operativni sistem, nalazi na nekoj od spoljnih memorija (tipično na hard ili SSD disku).

Master boot record

Punilac se nalazi u okviru **nultog sektora hard diska**, koji je poznat pod nazivom MBR (engl. *master boot record*). BIOS učitava program iz MBR-a u memoriju i predaje mu kontrolu.

Punilac

- Uloga punioca je da pronađe ostale komponente operativnog sistema na disku i učita ih u operativnu memoriju.

Punilac

- Uloga punioca je da pronađe ostale komponente operativnog sistema na disku i učita ih u operativnu memoriju.
- Punilac mora biti kompatibilan sa operativnim sistemom i mora biti upoznat sa strukturom operativnog sistema.

Punilac

- Uloga punioca je da pronađe ostale komponente operativnog sistema na disku i učita ih u operativnu memoriju.
- Punilac mora biti kompatibilan sa operativnim sistemom i mora biti upoznat sa strukturom operativnog sistema.
- Kod GNU/Linux – najpoznatiji su *Lilo* i *Grub*.

Punilac

- Uloga punioca je da pronađe ostale komponente operativnog sistema na disku i učita ih u operativnu memoriju.
- Punilac mora biti kompatibilan sa operativnim sistemom i mora biti upoznat sa strukturom operativnog sistema.
- Kod GNU/Linux – najpoznatiji su *Lilo* i *Grub*.
- Nakon što punilac učita jezgro operativnog sistema u operativnu memoriju i preda mu kontrolu, njegova uloga se završava.

Jezgro operativnog sistema – početni koraci

- Inicijalizuje svoje strukture podataka.

Jezgro operativnog sistema – početni koraci

- Inicijalizuje svoje strukture podataka.
- Inicijalizuje različite sistemske atribute: sistemskog vremena, vremenskog intervala tajmera i sl.

Jezgro operativnog sistema – početni koraci

- Inicijalizuje svoje strukture podataka.
- Inicijalizuje različite sistemske atribute: sistemskog vremena, vremenskog intervala tajmera i sl.
- Jezgro pokreće prvi korisnički program, čime se započinje proces inicijalizacije korisničkog okruženja i pokretanje sistemskih servisa.

Operativni sistem

Funkcije operativnog sistema

Operativni sistem

Funkcije operativnog sistema

Upravljanje procesorom

- Procesorsko vreme – najvažniji resurs (lako ako je jedan program)

Upravljanje procesorom

- Procesorsko vreme – najvažniji resurs (lako ako je jedan program)
- Prilikom izvršavanja programa najveći deo vremena će se tipično trošiti na interakciju sa drugim sporijim komponentama hardvera.

Upravljanje procesorom

- Procesorsko vreme – najvažniji resurs (lako ako je jedan program)
- Prilikom izvršavanja programa najveći deo vremena će se tipično trošiti na interakciju sa drugim sporijim komponentama hardvera.
- Dok program čeka na izvršenje spore ulazno-izlazne operacije, procesor stoji u mestu i ne radi ništa.

Upravljanje procesorom

- Procesorsko vreme – najvažniji resurs (lako ako je jedan program)
- Prilikom izvršavanja programa najveći deo vremena će se tipično trošiti na interakciju sa drugim sporijim komponentama hardvera.
- Dok program čeka na izvršenje spore ulazno-izlazne operacije, procesor stoji u mestu i ne radi ništa.
- Vreme izvršavanja ulazno-izlazne operacije iz ugla procesora traje veoma dugo: više hiljada instrukcija sporo...

Upravljanje procesorom – Multiprocesiranje

Multiprocesiranje

U memoriju se učita više programa.

Upravljanje procesorom – Multiprocesiranje

Multiprocesiranje

U memoriju se učita više programa.

U svakom trenutku, jedan program se izvršava na procesoru, dok ostali programi čekaju da dobiju procesor.

Upravljanje procesorom – Multiprocesiranje

Multiprocesiranje

U memoriju se učita više programa.

U svakom trenutku, jedan program se izvršava na procesoru, dok ostali programi čekaju da dobiju procesor.

> Kada program koji se izvršava mora da se zaustavi da bi sačekao neku sporu ulazno-izlaznu operaciju, operativni sistem predaje procesor na korišćenje drugog programu koji nastavlja svoj rad.

Danas – uobičajeno da više desetina programa se obavlja istovremeno.

Upravljanje procesorom – Multiprocesiranje

Multiprocesiranje

U memoriju se učita više programa.

U svakom trenutku, jedan program se izvršava na procesoru, dok ostali programi čekaju da dobiju procesor.

Kada program koji se izvršava mora da se zaustavi da bi sačekao neku sporu ulazno-izlaznu operaciju, operativni sistem predaje procesor na korišćenje drugog programu koji nastavlja svoj rad.

Upravljanje procesorom – Multiprocesiranje

Multiprocesiranje

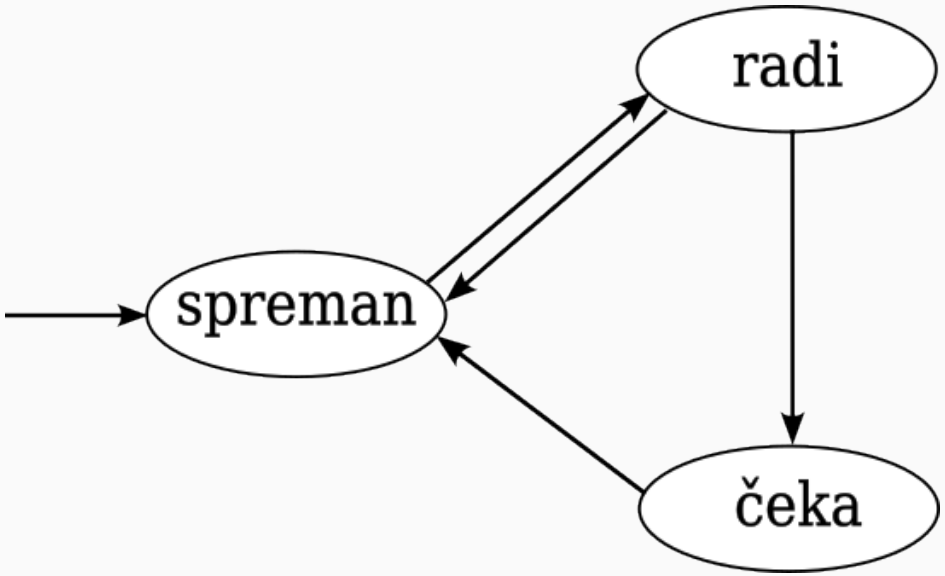
U memoriju se učita više programa.

U svakom trenutku, jedan program se izvršava na procesoru, dok ostali programi čekaju da dobiju procesor.

> Kada program koji se izvršava mora da se zaustavi da bi sačekao neku sporu ulazno-izlaznu operaciju, operativni sistem predaje procesor na korišćenje drugog programu koji nastavlja svoj rad.

Danas – uobičajeno da više desetina programa se obavlja istovremeno.

Raspoređivanje procesa



Slika 1: Graf stanja procesa

Raspoređivanje procesa

- Sa stanovišta efikasnosti upotrebe procesora, najbolja moguća strategija je ne prekidati rad procesa koji efektivno koristi procesor u punoj meri.

Raspoređivanje procesa

- Sa stanovišta efikasnosti upotrebe procesora, najbolja moguća strategija je ne prekidati rad procesa koji efektivno koristi procesor u punoj meri.
- Ipak, ovakva strategija može da ne bude fer prema drugim procesima koji čekaju na procesor.

Raspoređivanje procesa

- Sa stanovišta efikasnosti upotrebe procesora, najbolja moguća strategija je ne prekidati rad procesa koji efektivno koristi procesor u punoj meri.
- Ipak, ovakva strategija može da ne bude fer prema drugim procesima koji čekaju na procesor.
- U moderno vreme, korisnik želi da aktivno koristi više različitih programa i očekuje da ti programi brzo reaguju na njegove komande. Tj. potrebno je obezbediti korisniku utisak da se svi programi izvršavaju sve vreme, iako mi znamo da to nije moguće, s obzirom da imamo samo jedan procesor.

Raspoređivanje procesa

- Sa stanovišta efikasnosti upotrebe procesora, najbolja moguća strategija je ne prekidati rad procesa koji efektivno koristi procesor u punoj meri.
- Ipak, ovakva strategija može da ne bude fer prema drugim procesima koji čekaju na procesor.
- U moderno vreme, korisnik želi da aktivno koristi više različitih programa i očekuje da ti programi brzo reaguju na njegove komande. Tj. potrebno je obezbediti korisniku utisak da se svi programi izvršavaju sve vreme, iako mi znamo da to nije moguće, s obzirom da imamo samo jedan procesor.
- Ovo se postiže podelom vremena (engl. *time sharing*).

Raspoređivanje procesa

- Vremenski kvantum – svakom procesu odredi maksimalno vreme koje može da koristi u jednom krugu, pre nego što procesor prepusti drugom procesu (oko 50ms).

Raspoređivanje procesa

- Vremenski kvantum – svakom procesu odredi maksimalno vreme koje može da koristi u jednom krugu, pre nego što procesor prepusti drugom procesu (oko 50ms).
- Ukoliko proces koji je u stanju *radi* prekorači vremenski kvantum, a da pritom nije zahtevao ulazno-izlaznu operaciju, operativni sistem ga automatski prekida i prebacuje u stanje *spreman*.

Raspoređivanje procesa

- Vremenski kvantum – svakom procesu odredi maksimalno vreme koje može da koristi u jednom krugu, pre nego što procesor prepusti drugom procesu (oko 50ms).
- Ukoliko proces koji je u stanju *radi* prekorači vremenski kvantum, a da pritom nije zahtevao ulazno-izlaznu operaciju, operativni sistem ga automatski prekida i prebacuje u stanje *spreman*.
- Isovremeno, iz reda spremnih procesa bira drugi proces koga prebacuje u stanje *radi* i prepušta mu procesor na korišćenje.

Algoritama raspoređivanja procesa

- FCFS (engl. *first-come-first-served*)

Algoritama raspoređivanja procesa

- FCFS (engl. *first-come-first-served*)
- RR (engl. *round-robin*)

Algoritama raspoređivanja procesa

- FCFS (engl. *first-come-first-served*)
- RR (engl. *round-robin*)
- Algoritam zasnovan na *prioritetima*

Algoritama raspoređivanja procesa

- FCFS (engl. *first-come-first-served*)
- RR (engl. *round-robin*)
- Algoritam zasnovan na *prioritetima*
- SJF (engl. *shortest-job-first*)

Zamena konteksta

Zamena konteksta

Oduzimanje procesora jednom procesu i njegovo prepuštanje drugom procesu.

Zamena konteksta

Zamena konteksta

Oduzimanje procesora jednom procesu i njegovo prepuštanje drugom procesu.

Kontrolni blok procesa

Predstavlja deo struktura podataka operativnog sistema gde se upisuju vrednosti registara procesora u trenutku prekida rada procesa.

Svaki proces ima svoj kontrolni blok.

Operativni sistem

Upravljanje operativnom memorijom

Upravljanje operativnom memorijom

- **Prvi problem** koji nastaje u slučaju koegzistencije više programa u memoriji je mogućnost da jedan proces slučajno ili namerno pristupi podacima drugog procesa ili samog operativnog sistema.

Upravljanje operativnom memorijom

- **Prvi problem** koji nastaje u slučaju koegzistencije više programa u memoriji je mogućnost da jedan proces slučajno ili namerno pristupi podacima drugog procesa ili samog operativnog sistema.
- Da bi to sprečili potrebno je da na neki način izolujemo memorijske prostore pojedinačnih procesa i sprečimo adresiranje memorijskih lokacija izvan tog prostora.

Virtuelna memorija

Virtuelni adresni prostor

Fiktivni adresni prostor koji ne postoji zaista, već se njegova egzistencija emulira od strane procesora, u saradnji sa operativnim sistemom.

Virtuelna memorija

Virtuelni adresni prostor

Fiktivni adresni prostor koji ne postoji zaista, već se njegova egzistencija emulira od strane procesora, u saradnji sa operativnim sistemom.

On sadrži kompletan mašinski program, kao i sve podatke koje proces koristi tokom svog rada.

Virtuelna memorija

Virtuelni adresni prostor

Fiktivni adresni prostor koji ne postoji zaista, već se njegova egzistencija emulira od strane procesora, u saradnji sa operativnim sistemom.

On sadrži kompletan mašinski program, kao i sve podatke koje proces koristi tokom svog rada.

Virtuelni adresni prostor je jedini prostor koji je vidljiv procesu tokom izvršavanja.

Virtuelna memorija

Virtuelni adresni prostor

Fiktivni adresni prostor koji ne postoji zaista, već se njegova egzistencija emulira od strane procesora, u saradnji sa operativnim sistemom.

On sadrži kompletan mašinski program, kao i sve podatke koje proces koristi tokom svog rada.

Virtuelni adresni prostor je jedini prostor koji je vidljiv procesu tokom izvršavanja.

Virtuelna memorija

Sve instrukcije programa referišu isključivo na adrese iz tog virtuelnog adresnog prostora.

Virtuelna memorija

Sve instrukcije programa referišu isključivo na adrese iz tog virtuelnog adresnog prostora.

Adresa tekuće instrukcije programa koja se nalazi u programskom brojaču procesora je takođe adresa iz tog virtuelnog prostora.

> Adrese lokacija unutar virtuelnog adresnog prostora nazivaju se virtuelne adrese.

Fizički adresni prostor

- Fizička RAM memorija koja je instalirana na računaru.

Fizički adresni prostor

- Fizička RAM memorija koja je instalirana na računaru.
- Fizički prostor je jedinstven i zajednički za sve procese.

Fizički adresni prostor

- Fizička RAM memorija koja je instalirana na računaru.
- Fizički prostor je jedinstven i zajednički za sve procese.
- Virtuelnim adresama svakog procesa pridružujemo fizičke adrese u RAM-u.

Kada proces želi da pristupi nekoj adresi u svom virtuelnom prostoru, ta adresa se propušta kroz deo procesora koji se zove memorijska jedinica (engl. *memory unit*) koja uz pomoć informacija koje je unapred obezbedio operativni sistem automatski prevodi virtuelnu adresu u fizičku adresu na kojoj se traženi podatak zaista nalazi, a zatim procesor pristupa toj fizičkoj adresi u RAM memoriji.

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Parcijalno učitavanje

Delovi virtuelnog adresnog prostora nekog procesa su „učitani" u fizičku memoriju, dok se ostali delovi virtuelnog prostora mogu čuvati u spoljnoj memoriji (tipično na hard ili SSD disku).

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Parcijalno učitavanje

Delovi virtuelnog adresnog prostora nekog procesa su „učitani" u fizičku memoriju, dok se ostali delovi virtuelnog prostora mogu čuvati u spoljnoj memoriji (tipično na hard ili SSD disku).

- Kada proces pristupi nekoj virtuelnoj adresi za koju se ispostavi da nema pridruženu fizičku adresu, dolazi do automatskog prekida rada programa i predaje kontrole operativnom sistemu.

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Parcijalno učitavanje

Delovi virtuelnog adresnog prostora nekog procesa su „učitani" u fizičku memoriju, dok se ostali delovi virtuelnog prostora mogu čuvati u spoljnoj memoriji (tipično na hard ili SSD disku).

- Kada proces pristupi nekoj virtuelnoj adresi za koju se ispostavi da nema pridruženu fizičku adresu, dolazi do automatskog prekida rada programa i predaje kontrole operativnom sistemu.
- Operativni sistem razrešava ovu situaciju tako što sa hard diska učita deo virtuelnog prostora procesa koji sadrži traženu adresu u neki slobodni deo operativne memorije.

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Parcijalno učitavanje

Delovi virtuelnog adresnog prostora nekog procesa su „učitani" u fizičku memoriju, dok se ostali delovi virtuelnog prostora mogu čuvati u spoljnoj memoriji (tipično na hard ili SSD disku).

- Kada proces pristupi nekoj virtuelnoj adresi za koju se ispostavi da nema pridruženu fizičku adresu, dolazi do automatskog prekida rada programa i predaje kontrole operativnom sistemu.
- Operativni sistem razrešava ovu situaciju tako što sa hard diska učita deo virtuelnog prostora procesa koji sadrži traženu adresu u neki slobodni deo operativne memorije.
- Nakon toga operativni sistem vraća kontrolu prekinutom

Upravljanje operativnom memorijom

- Još jedan problem koji može nastati prilikom upravljanja memorijom jeste problem nedostatka prostora u fizičkoj memoriji.

Parcijalno učitavanje

Delovi virtuelnog adresnog prostora nekog procesa su „učitani" u fizičku memoriju, dok se ostali delovi virtuelnog prostora mogu čuvati u spoljnoj memoriji (tipično na hard ili SSD disku).

- Kada proces pristupi nekoj virtuelnoj adresi za koju se ispostavi da nema pridruženu fizičku adresu, dolazi do automatskog prekida rada programa i predaje kontrole operativnom sistemu.
- Operativni sistem razrešava ovu situaciju tako što sa hard diska učita deo virtuelnog prostora procesa koji sadrži traženu adresu u neki slobodni deo operativne memorije.
- Nakon toga operativni sistem vraća kontrolu prekinutom

Upravljanje operativnom memorijom

Ukupna veličina svih virtuelnih adresnih prostora svih aktivnih procesa je obično značajno veća od veličine fizičke memorije računara.

Upravljanje operativnom memorijom

Ukupna veličina svih virtuelnih adresnih prostora svih aktivnih procesa je obično značajno veća od veličine fizičke memorije računara.

Problem: Preveliki broj aktivnih procesa može značajno pogoršati performanse računara, ukoliko to dovede do prepunjenosti fizičke memorije.

Dodeljivanje fizičke memorije procesu – Kontinualno

- Kontinualno – svakom procesu se dodeljuje odgovarajući broj **susednih** adresa fizičke memorije

Dodeljivanje fizičke memorije procesu – Kontinualno

- Kontinualno – svakom procesu se dodeljuje odgovarajući broj **susednih** adresa fizičke memorije
- Prednost – implementacija virtuelne memorije u slučaju kontinualnog dodeljivanja fizičke memorije je veoma jednostavna.

Dodeljivanje fizičke memorije procesu – Kontinualno

- Nedostaci – parcijalno učitavanje virtuelnog prostora procesa u cilju uštede fizičke memorije biti otežano, jer će različiti delovi virtuelnog adresnog prostora koji se zasebno po potrebi učitavaju morati da budu učitavani u susedne zone u memoriji, što često neće biti moguće jednostavno realizovati.

Dodeljivanje fizičke memorije procesu – Kontinualno

- Nedostaci – parcijalno učitavanje virtuelnog prostora procesa u cilju uštede fizičke memorije biti otežano, jer će različiti delovi virtuelnog adresnog prostora koji se zasebno po potrebi učitavaju morati da budu učitavani u susedne zone u memoriji, što često neće biti moguće jednostavno realizovati.
- Nedostaci – pojava fragmentacije fizičke memorije. Operativni sistem će slobodan prostor moći da dodeli nekom novom procesu samo ako taj novi proces ne zahteva više memorije. može lako dogoditi da za neki novi proces mi imamo dovoljno slobodnog prostora posmatrano ukupno, ali ga nemamo „u komadu”.

Dodeljivanje fizičke memorije procesu – Nekontinualno

- Nije neohodno da uzastopne virtuelne adrese budu preslikane u uzastopne fizičke adrese.

Dodeljivanje fizičke memorije procesu – Nekontinualno

- Nije neohodno da uzastopne virtuelne adrese budu preslikane u uzastopne fizičke adrese.
- Virtuelno i fizičku memoriju izdelimo na zone i delove.

Dodeljivanje fizičke memorije procesu – Nekontinualno

- Nije neohodno da uzastopne virtuelne adrese budu preslikane u uzastopne fizičke adrese.
- Virtuelno i fizičku memoriju izdelimo na zone i delove.
- Nedostatak – znatno komplikovanije za implementaciju. Memorijska jedinica znatno kompleksnija.

Upravljanje memorijom – straničenje

Straničenju (engl. *paging*) – sistem virtuelne memorije zasnovan na nekontinualnom dodeljivanju fizičkih adresa.

Svi procesi imaju virtuelni adresni prostor iste veličine.

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32}\,GB$ prostora.

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32}\,GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32} GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.
- Neka su sve stranice veličine 4KB ($= 2^{12}$ bajtova)

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32} GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.
- Neka su sve stranice veličine 4KB ($= 2^{12}$ bajtova)
- To znači da se ceo virtuelni prostor procesa sastoji iz 2^{20} stranica od po 2^{12} bajtova.

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32} GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.
- Neka su sve stranice veličine 4KB ($= 2^{12}$ bajtova)
- To znači da se ceo virtuelni prostor procesa sastoji iz 2^{20} stranica od po 2^{12} bajtova.
- 32-bitna virtuelna adresa (*va*) sastoji iz dva dela:

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32} GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.
- Neka su sve stranice veličine 4KB ($= 2^{12}$ bajtova)
- To znači da se ceo virtuelni prostor procesa sastoji iz 2^{20} stranica od po 2^{12} bajtova.
- 32-bitna virtuelna adresa (*va*) sastoji iz dva dela:
  - viših 20 bitova predstavljaju redni broj stranice (označimo ga sa *ha*)

Upravljanje memorijom – straničenje – primer

- Pretpostavimo da imamo 32-bitni virtuelni adresni prostor.
- Imamo 2^{32} različitih adresa. A to znači $2^{32} GB$ prostora.
- Virtuelni prostor procesa se logički deli na *stranice*.
- Neka su sve stranice veličine 4KB ($= 2^{12}$ bajtova)
- To znači da se ceo virtuelni prostor procesa sastoji iz 2^{20} stranica od po 2^{12} bajtova.
- 32-bitna virtuelna adresa (*va*) sastoji iz dva dela:
  - viših 20 bitova predstavljaju redni broj stranice (označimo ga sa *ha*)
  - nižih 12 bitova (označimo ih sa *la*) predstavljaju *pomeraj* (engl. *offset*) u okviru stranice

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)
- Nazivamo ih okviri stranica (engl. *page frames*).

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)
- Nazivamo ih okviri stranica (engl. *page frames*).
- Pretpostavimo da imamo 16GB (2^{34}) fizičkog adresnog prostora.

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)
- Nazivamo ih okviri stranica (engl. *page frames*).
- Pretpostavimo da imamo 16GB (2^{34}) fizičkog adresnog prostora.
- Ovaj prostor se deli na 2^{22} okvira od po 2^{12} bajtova.

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)
- Nazivamo ih okviri stranica (engl. *page frames*).
- Pretpostavimo da imamo 16GB (2^{34}) fizičkog adresnog prostora.
- Ovaj prostor se deli na 2^{22} okvira od po 2^{12} bajtova.
- U svaki okvir može biti učitana jedna stranica virtuelnog adresnog prostora nekog od aktivnih procesa.

Upravljanje memorijom – straničenje – primer

- Fizički adresni prostor se takođe deli na delove koji su jednake veličine kao i stranice (u našem primeru 4KB)
- Nazivamo ih okviri stranica (engl. *page frames*).
- Pretpostavimo da imamo 16GB (2^{34}) fizičkog adresnog prostora.
- Ovaj prostor se deli na 2^{22} okvira od po 2^{12} bajtova.
- U svaki okvir može biti učitana jedna stranica virtuelnog adresnog prostora nekog od aktivnih procesa.
- Ne postoji zahtev da susedne stranice u virtuelnom adresnom prostoru nekog procesa budu u susednim okvirima fizičkog adresnog prostora.

Upravljanje memorijom – straničenje

Tabela stranica (engl. *page table*)

Za svaki proces tabela sadrži po jednu stavku za svaku od stranica procesa.

Stavke su jednake veličine (npr. 8 bajta) i nalaze se jedna za drugom u memoriji, počev od neke fizičke adrese koja je poznata operativnom sistemu.

Upravljanje memorijom – straničenje

Tabela stranica (engl. *page table*)

Za svaki proces tabela sadrži po jednu stavku za svaku od stranica procesa.
Stavke su jednake veličine (npr. 8 bajta) i nalaze se jedna za drugom u memoriji, počev od neke fizičke adrese koja je poznata operativnom sistemu.

U našem primeru, za svaki pokrenuti proces će postojati podatak o 2^{20} svakog procesa.

Upravljanje memorijom – straničenje

Tabela stranica (engl. *page table*)

Za svaki proces tabela sadrži po jednu stavku za svaku od stranica procesa.
Stavke su jednake veličine (npr. 8 bajta) i nalaze se jedna za drugom u memoriji, počev od neke fizičke adrese koja je poznata operativnom sistemu.

U našem primeru, za svaki pokrenuti proces će postojati podatak o 2^{20} svakog procesa.

Svaka stavka sadrži, između ostalog, informaciju o tome da li je odgovarajuća stranica prisutna u fizičkoj memoriji i, ako jeste, u kom okviru.

Upravljanje memorijom – straničenje

Registar PTA

PTA - page table address) registar — sadrži *fizičku* adresu tabele stranica za proces koji se trenutno izvršava u procesoru.

Kako se adresa dobija ako je stranica u memoriji:

- Na osnovu višeg dela virtuelne adrese *ha* odredi redni broj stranice virtuelnog adresnog prostora.

Upravljanje memorijom – straničenje

Registar PTA

PTA - page table address) registar — sadrži *fizičku* adresu tabele stranica za proces koji se trenutno izvršava u procesoru.

Kako se adresa dobija ako je stranica u memoriji:

- Na osnovu višeg dela virtuelne adrese *ha* odredi redni broj stranice virtuelnog adresnog prostora.
- Pomoću *PTA* registra pristupa stavci tabele stranica sa tim rednim brojem.

Upravljanje memorijom – straničenje

Registar PTA

PTA - page table address) registar — sadrži *fizičku* adresu tabele stranica za proces koji se trenutno izvršava u procesoru.

Kako se adresa dobija ako je stranica u memoriji:

- Na osnovu višeg dela virtuelne adrese *ha* odredi redni broj stranice virtuelnog adresnog prostora.
- Pomoću *PTA* registra pristupa stavci tabele stranica sa tim rednim brojem.
- Ova stavka se učitava u memorijsku jedinicu i iz nje se utvrđuje da li je stranica učitana u fizičku memoriju ili ne.

Upravljanje memorijom – straničenje

Registar PTA

PTA - page table address) registar — sadrži *fizičku* adresu tabele stranica za proces koji se trenutno izvršava u procesoru.

Kako se adresa dobija ako je stranica u memoriji:

- Na osnovu višeg dela virtuelne adrese *ha* odredi redni broj stranice virtuelnog adresnog prostora.
- Pomoću *PTA* registra pristupa stavci tabele stranica sa tim rednim brojem.
- Ova stavka se učitava u memorijsku jedinicu i iz nje se utvrđuje da li je stranica učitana u fizičku memoriju ili ne.
- Tada se iz stavke pročita 22-bitni redni broj okvira u fizičkoj memoriji gde se stranica nalazi.

Upravljanje memorijom – straničenje

Registar PTA

PTA - page table address) registar — sadrži *fizičku* adresu tabele stranica za proces koji se trenutno izvršava u procesoru.

Kako se adresa dobija ako je stranica u memoriji:

- Na osnovu višeg dela virtuelne adrese *ha* odredi redni broj stranice virtuelnog adresnog prostora.
- Pomoću *PTA* registra pristupa stavci tabele stranica sa tim rednim brojem.
- Ova stavka se učitava u memorijsku jedinicu i iz nje se utvrđuje da li je stranica učitana u fizičku memoriju ili ne.
- Tada se iz stavke pročita 22-bitni redni broj okvira u fizičkoj memoriji gde se stranica nalazi.
- Na ovo 22-bitno polje se dopisuje niži deo virtuelne adrese *la*, čime se dobija 34-bitna fizička adresa podatka u memoriji.

Upravljanje memorijom – straničenje

Ukoliko se ispostavi da stranica nije učitana u fizičku memoriju:

- Dolazi do prekida rada programa – greška straničenja, engl. *page fault*.

Upravljanje memorijom – straničenje

Ukoliko se ispostavi da stranica nije učitana u fizičku memoriju:

- Dolazi do prekida rada programa – greška straničenja, engl. *page fault*.
- Kontrola se predaje operativnom sistemu koji sa hard diska učitava traženu stranicu u neki slobodan okvir u fizičkoj memoriji.

Upravljanje memorijom – straničenje

Ukoliko se ispostavi da stranica nije učitana u fizičku memoriju:

- Dolazi do prekida rada programa – greška straničenja, engl. *page fault*.
- Kontrola se predaje operativnom sistemu koji sa hard diska učitava traženu stranicu u neki slobodan okvir u fizičkoj memoriji.
- Ažurira se odgovarajuća stavka u tabeli stranica.

Upravljanje memorijom – straničenje

Ukoliko se ispostavi da stranica nije učitana u fizičku memoriju:

- Dolazi do prekida rada programa – greška straničenja, engl. *page fault*.
- Kontrola se predaje operativnom sistemu koji sa hard diska učitava traženu stranicu u neki slobodan okvir u fizičkoj memoriji.
- Ažurira se odgovarajuća stavka u tabeli stranica.
- Kontrola se vraća procesu koji ponovo pokušava da izvrši istu instrukciju.

Upravljanje memorijom – straničenje

Ukoliko se ispostavi da stranica nije učitana u fizičku memoriju:

- Dolazi do prekida rada programa – greška straničenja, engl. *page fault*.
- Kontrola se predaje operativnom sistemu koji sa hard diska učitava traženu stranicu u neki slobodan okvir u fizičkoj memoriji.
- Ažurira se odgovarajuća stavka u tabeli stranica.
- Kontrola se vraća procesu koji ponovo pokušava da izvrši istu instrukciju.
- Ako prilikom učitavanja nove stranice nema slobodnih okvira, tada operativni sistem bira „žrtvu" – stranicu koju će izbaciti iz svog okvira i iskopirati je na hard disk, kako bi se okvir oslobodio za novu stranicu.
  Ovo nije nimalo trivijalno – zahteva anticipaciju budućeg korišćenja stranica.

Operativni sistem

Upravljanje ulazno-izlaznim uređajima

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni uređaji: tastatura, miš, monitor, ekrani osetljivi na dodir, štampači, skeneri, zvučnika, mikrofona, kamera, mrežni uređaji, spoljne memorije i sl.

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni uređaji: tastatura, miš, monitor, ekrani osetljivi na dodir, štampači, skeneri, zvučnika, mikrofona, kamera, mrežni uređaji, spoljne memorije i sl.

Procesor računara komunicira sa ulazno-izlaznim uređajima podsredstvom ulazno-izlaznih kontrolera

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni uređaji: tastatura, miš, monitor, ekrani osetljivi na dodir, štampači, skeneri, zvučnika, mikrofona, kamera, mrežni uređaji, spoljne memorije i sl.

> Procesor računara komunicira sa ulazno-izlaznim uređajima podsredstvom ulazno-izlaznih kontrolera

Kontroler se povezuje na magistralu i procesor i sa njima komunicira na veoma sličan način kao i sa operativnom memorijom, razmenjujući *podatke*, *komande* i *statusne informacije* putem magistrale.

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni podsistem – deo operativnog sistema zadužen za komunikacija sa ulazno-izlaznim uređajima.

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni podsistem – deo operativnog sistema zadužen za komunikacija sa ulazno-izlaznim uređajima.

Ovaj sistem se sastoji iz dva sloja:

- Na **nižem sloju** nalaze se upravljački programi zaduženi za direktnu komunikaciju sa kontrolerima. Ovi programi su poznati i kao drajveri (engl. *device drivers*) i u njima su implementirani svi detalji komandnog jezika konkretnog uređaja.

Upravljanje ulazno-izlaznim uređajima

Ulazno-izlazni podsistem – deo operativnog sistema zadužen za komunikacija sa ulazno-izlaznim uređajima.

Ovaj sistem se sastoji iz dva sloja:

- Na **nižem sloju** nalaze se upravljački programi zaduženi za direktnu komunikaciju sa kontrolerima. Ovi programi su poznati i kao drajveri (engl. *device drivers*) i u njima su implementirani svi detalji komandnog jezika konkretnog uređaja.
- Na **višem sloju** implementiran je interfejs ka drugim delovima operativnog sistema i korisničkim programima, a koji je nezavisan od konkretnog tipa i modela uređaja. Ovim interfejsom se obezbeđuje apstrakcija hardvera,

Upravljanje ulazno-izlaznim uređajima – baferi

Bafer

Bafer predstavlja strukturu podataka koja može skladištiti podatke koji se prenose između dve strane u komunikaciji – u ovom slučaju između ulazno-izlaznog kontrolera i operativnog sistema.

Upravljanje ulazno-izlaznim uređajima – baferi

Bafer

Bafer predstavlja strukturu podataka koja može skladištiti podatke koji se prenose između dve strane u komunikaciji – u ovom slučaju između ulazno-izlaznog kontrolera i operativnog sistema.

Uloga bafera je da amortizuje razliku u brzini između dve strane koje komuniciraju.

Operativni sistem

Upravljanje podacima

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).
- Računar ove uređaje vidi kao ulazno-izlazne uređaje i sa njima komunicira putem odgovarajućih kontrolera.

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).
- Računar ove uređaje vidi kao ulazno-izlazne uređaje i sa njima komunicira putem odgovarajućih kontrolera.
    - SATA kontroler se koristi za komunikaciju sa hard i SSD diskovima

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).
- Računar ove uređaje vidi kao ulazno-izlazne uređaje i sa njima komunicira putem odgovarajućih kontrolera.
  - SATA kontroler se koristi za komunikaciju sa hard i SSD diskovima
  - USB kontroler koristi za komunikaciju sa USB fleš memorijama

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).
- Računar ove uređaje vidi kao ulazno-izlazne uređaje i sa njima komunicira putem odgovarajućih kontrolera.
  - SATA kontroler se koristi za komunikaciju sa hard i SSD diskovima
  - USB kontroler koristi za komunikaciju sa USB fleš memorijama
- Svaka spoljna memorija se može logički posmatrati kao niz adresibilnih lokacija fiksirane veličine. Na primer, hard diskovi se mogu posmatrati kao nizovi *sektora*, tipične veličine 512 bajtova.

Upravljanje podacima

- Podaci se na računaru skladište u spoljnim memorijama (poput hard i SSD diskova, fleš memorija i sl.).
- Računar ove uređaje vidi kao ulazno-izlazne uređaje i sa njima komunicira putem odgovarajućih kontrolera.
  - SATA kontroler se koristi za komunikaciju sa hard i SSD diskovima
  - USB kontroler koristi za komunikaciju sa USB fleš memorijama
- Svaka spoljna memorija se može logički posmatrati kao niz adresibilnih lokacija fiksirane veličine. Na primer, hard diskovi se mogu posmatrati kao nizovi *sektora*, tipične veličine 512 bajtova.
-

Upravljanje podacima – sistem datoteka (engl. *file system*)

Pod *datotekom* ili *fajlom* (engl. *file*) podrazumevamo skup logički povezanih podataka koji čine jednu celinu.

Na primer: tekstualni dokument, slika, video zapis, ali i računarski program.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Pod *datotekom* ili *fajlom* (engl. *file*) podrazumevamo skup logički povezanih podataka koji čine jednu celinu.

Na primer: tekstualni dokument, slika, video zapis, ali i računarski program.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Fajlovi mogu imati:

- svoje ime koje ga identifikuje;

Na nekim fajl sistemima, tip fajla može biti određen *ekstenzijom*, tj. specifičnim nastavkom imena (npr. *.txt*, *.doc*, *.jpg*, *.exe*, i sl.), dok na drugim sistemima naziv fajla nema uticaja na tip, već se tip prepoznaje na drugačiji način.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Fajlovi mogu imati:

- svoje ime koje ga identifikuje;
- sadržaj koji je predstavljen nizom bajtova, u formatu koji zavisi od specifičnog tipa fajla;

Na nekim fajl sistemima, tip fajla može biti određen *ekstenzijom*, tj. specifičnim nastavkom imena (npr. *.txt*, *.doc*, *.jpg*, *.exe*, i sl.), dok na drugim sistemima naziv fajla nema uticaja na tip, već se tip prepoznaje na drugačiji način.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Fajlovi mogu imati:

- svoje ime koje ga identifikuje;
- sadržaj koji je predstavljen nizom bajtova, u formatu koji zavisi od specifičnog tipa fajla;
- veličine (u bajtovima);

Na nekim fajl sistemima, tip fajla može biti određen *ekstenzijom*, tj. specifičnim nastavkom imena (npr. *.txt*, *.doc*, *.jpg*, *.exe*, i sl.), dok na drugim sistemima naziv fajla nema uticaja na tip, već se tip prepoznaje na drugačiji način.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Fajlovi mogu imati:

- svoje ime koje ga identifikuje;
- sadržaj koji je predstavljen nizom bajtova, u formatu koji zavisi od specifičnog tipa fajla;
- veličine (u bajtovima);
- vlasništvo, prava pristupa;

Na nekim fajl sistemima, tip fajla može biti određen *ekstenzijom*, tj. specifičnim nastavkom imena (npr. *.txt*, *.doc*, *.jpg*, *.exe*, i sl.), dok na drugim sistemima naziv fajla nema uticaja na tip, već se tip prepoznaje na drugačiji način.

Upravljanje podacima – sistem datoteka (engl. *file system*)

Fajlovi mogu imati:

- svoje ime koje ga identifikuje;
- sadržaj koji je predstavljen nizom bajtova, u formatu koji zavisi od specifičnog tipa fajla;
- veličine (u bajtovima);
- vlasništvo, prava pristupa;
- datuma poslednje modifikacije i sl.

Na nekim fajl sistemima, tip fajla može biti određen *ekstenzijom*, tj. specifičnim nastavkom imena (npr. *.txt*, *.doc*, *.jpg*, *.exe*, i sl.), dok na drugim sistemima naziv fajla nema uticaja na tip, već se tip prepoznaje na drugačiji način.

Upravljanje podacima – sistem datoteka (engl. *file system*)

- Sistem datoteka obično podrazumeva dva aspekta.

Upravljanje podacima – sistem datoteka (engl. *file system*)

- Sistem datoteka obično podrazumeva dva aspekta.
- Logički aspekt predstavlja način na koji se podaci koji se nalaze skladišteni u okviru sistema datoteka predstavljaju korisniku.

Upravljanje podacima – sistem datoteka (engl. *file system*)

- Sistem datoteka obično podrazumeva dva aspekta.
- Logički aspekt predstavlja način na koji se podaci koji se nalaze skladišteni u okviru sistema datoteka predstavljaju korisniku.
- Fizički aspekt podrazumeva na koji način se podaci fizički skladište u spoljnoj memoriji.

Upravljanje podacima – sistem datoteka – Logički aspekt

- Danas su uobičajeni hijerarhijski sistemi datoteka, kod kojih se fajlovi organizuju u direktorijume ili fascikle.

Upravljanje podacima – sistem datoteka – Logički aspekt

- Danas su uobičajeni hijerarhijski sistemi datoteka, kod kojih se fajlovi organizuju u direktorijume ili fascikle.
- Svaki direktorijum može sadržati fajlove, kao i druge direktorijume(koje nazivamo poddirektorijumi).

Upravljanje podacima – sistem datoteka – Logički aspekt

- Danas su uobičajeni hijerarhijski sistemi datoteka, kod kojih se fajlovi organizuju u direktorijume ili fascikle.
- Svaki direktorijum može sadržati fajlove, kao i druge direktorijume(koje nazivamo poddirektorijumi).
- Na vrhu hijerarhije nalazi se koreni direktorijum (engl. *root directory*).

Upravljanje podacima – sistem datoteka – Fizički aspekt

- Memorijski prostor spoljne memorije može posmatrati kao niz adresibilnih jedinica.

Upravljanje podacima – sistem datoteka – Fizički aspekt

- Memorijski prostor spoljne memorije može posmatrati kao niz adresibilnih jedinica.
- Na **početku tog** memorijskog prostora skladišti:

Upravljanje podacima – sistem datoteka – Fizički aspekt

- Memorijski prostor spoljne memorije može posmatrati kao niz adresibilnih jedinica.
- Na **početku tog** memorijskog prostora skladišti:
  - meta-informacije o sebi (tip sistema datoteka, verzija, i sl.);

Upravljanje podacima – sistem datoteka – Fizički aspekt

- Memorijski prostor spoljne memorije može posmatrati kao niz adresibilnih jedinica.
- Na **početku tog** memorijskog prostora skladišti:
  - meta-informacije o sebi (tip sistema datoteka, verzija, i sl.);
  - strukture podataka koje sadrže informacije o fajlovima i direktorijumima i omogućavaju pronalaženje sadržaja konkretnih fajlova u okviru fajl sistema.

Upravljanje podacima – sistem datoteka – tipovi

Svaki operativni sistem podržava određeni skup tipova sistema datoteka i može pristupati samo sistemima datoteka tih tipova.

- Sistemi FAT i NTFS – *DOS* i *Windows*. Podržani i od strane *GNU/Linux* operativnog sistema.

Upravljanje podacima – sistem datoteka – tipovi

Svaki operativni sistem podržava određeni skup tipova sistema datoteka i može pristupati samo sistemima datoteka tih tipova.

- Sistemi FAT i NTFS – *DOS* i *Windows*. Podržani i od strane *GNU/Linux* operativnog sistema.
- Sistemi ext4 i ReiserFS – *Linux*.

Upravljanje podacima – sistem datoteka – tipovi

Svaki operativni sistem podržava određeni skup tipova sistema datoteka i može pristupati samo sistemima datoteka tih tipova.

- Sistemi FAT i NTFS – *DOS* i *Windows*. Podržani i od strane *GNU/Linux* operativnog sistema.
- Sistemi ext4 i ReiserFS – *Linux*.

Upravljanje podacima – sistem datoteka – tipovi

Svaki operativni sistem podržava određeni skup tipova sistema datoteka i može pristupati samo sistemima datoteka tih tipova.

- Sistemi FAT i NTFS – *DOS* i *Windows*. Podržani i od strane *GNU/Linux* operativnog sistema.
- Sistemi ext4 i ReiserFS – *Linux*.

Postupak kreiranja sistema datoteka poznat je i kao *formatiranje*.

Upravljanje podacima – sistem datoteka – tipovi

Svaki operativni sistem podržava određeni skup tipova sistema datoteka i može pristupati samo sistemima datoteka tih tipova.

- Sistemi FAT i NTFS – *DOS* i *Windows*. Podržani i od strane *GNU/Linux* operativnog sistema.
- Sistemi ext4 i ReiserFS – *Linux*.

Postupak kreiranja sistema datoteka poznat je i kao *formatiranje*.

Nakon formatiranja, korisnik dobija pristup sistemu datoteka koji je inicijalno prazan.

Upravljanje podacima – sistem datoteka – particije

–

Disk se može logički podeliti na particije – međusobno disjunktne celine, pri čemu se svaka particija sastoji iz skupa susednih sektora.

–

Disk se može logički podeliti na particije – međusobno disjunktne celine, pri čemu se svaka particija sastoji iz skupa susednih sektora.

Informacije o particionisanju (tj. od kog do kog sektora se prostire koja particija) se obično nalaze u početnim sektorima diska koji nisu deo ni jedne particije.

–

Disk se može logički podeliti na particije – međusobno disjunktne celine, pri čemu se svaka particija sastoji iz skupa susednih sektora.

Informacije o particionisanju (tj. od kog do kog sektora se prostire koja particija) se obično nalaze u početnim sektorima diska koji nisu deo ni jedne particije.

Sada se sistemi datoteka mogu nezavisno kreirati u svakoj od particija.

Sistem datoteka – relativna i apsolutna putanja

Lokacija svakog fajla ili direktorijuma u okviru hijerarhijskog sistema datoteka se opisuje putanjom.

Sistem datoteka – relativna i apsolutna putanja

Lokacija svakog fajla ili direktorijuma u okviru hijerarhijskog sistema datoteka se opisuje putanjom.

Apsolutna putanja

Apsolutna putanja se dobija navođenjem svih direktorijuma u stablu prateći put od korenog direktorijuma do datog fajla ili direktorijuma.

Na primer: `/home/user/Downloads/Pictures/sunce.jpg`

Sistem datoteka – relativna i apsolutna putanja

Lokacija svakog fajla ili direktorijuma u okviru hijerarhijskog sistema datoteka se opisuje putanjom.

Apsolutna putanja

Apsolutna putanja se dobija navođenjem svih direktorijuma u stablu prateći put od korenog direktorijuma do datog fajla ili direktorijuma.

Na primer: `/home/user/Downloads/Pictures/sunce.jpg`

Relativna putanja

Relativna putanja predstavlja putanju nekog fajla ili direktorijuma u odnosu na neki drugi direktorijum u stablu direktorijuma (tipično u odnosu na *tekući direktorijum*, tj. radni direktorijum koji je pridružen procesu koji pristupa sistemu datoteka).

Na primer: ako je tekući direktorijum `Downloads`, relativna putanja

Operativni sistem

Komunikacija između procesa

Komunikacija između procesa

Najjednostavniji mehanizmi podrazumevaju korišćenje signala ka operatvnom sistemu.

Komunikacija između procesa

Najjednostavniji mehanizmi podrazumevaju korišćenje signala ka operatvnom sistemu.

Složeniji mehanizmi komunikacije omogućavaju razmenu proizvoljnih podataka putem kreiranog kanala (primer takvih mehanizama su cevi (engl. *pipe*) i priključci (engl. *socket*) na *UNIX* sistemima).

Komunikacija između procesa

Najjednostavniji mehanizmi podrazumevaju korišćenje signala ka operatvnom sistemu.

Složeniji mehanizmi komunikacije omogućavaju razmenu proizvoljnih podataka putem kreiranog kanala (primer takvih mehanizama su cevi (engl. *pipe*) i priključci (engl. *socket*) na *UNIX* sistemima).

Deljena memorija

najefikasniji način – deljena memorija (engl. *shared memory*). Ovaj način komunikacije podrazumeva da dva procesa dele neki zajednički deo virtuelnog adresnog prostora.

Operativni sistem

Komunikacija sa korisnikom

Komunikacija sa korisnikom

Svaki operativni sistem sadrži komponentu koja se zove korisnički interfejs.

Komunikacija sa korisnikom

Svaki operativni sistem sadrži komponentu koja se zove korisnički interfejs.

Korisnički interfejs predstavlja skup uslužnih programa koji su povezani sa standardnim ulazno-izlaznim uređajima poput tastature, miša i monitora, preko kojih mogu komunicirati sa korisnikom.

Korisnik može korisničkom interfejsu izdavati komande kojima od operativnog sistema zahteva neku akciju.

Korisnički interfejs

- Prvobitno, korisnički interfejs je bio tekstualni.

Korisnički interfejs

- Prvobitno, korisnički interfejs je bio tekstualni.
- To je podrazumevalo da korisnik poznaje sintaksu komandnog interfejsa.

Korisnički interfejs

- Prvobitno, korisnički interfejs je bio tekstualni.
- To je podrazumevalo da korisnik poznaje sintaksu komandnog interfejsa.

> Grafički korisnički interfejs (engl. *graphical user interface* (GUI)) – korisnik izdaje akcije pomoću miša, pritiskom na različite grafičke elemente koji su prikazani na ekranu(poput dugmadi, menija, ikona i sl.).

- Korisnički intefejs korisniku takođe saopštavati poruke koristeći grafičke elemente.

Operativni sistem

Hardverska podrška operativnim sistemima

Operativni sistem

Nivoi privilegija procesora

Nivoi privilegija procesora

Kako bi se fizički sprečilo da korisnički programi direktno pristupaju resursima koji im nisu namenjeni, procesor obično podržava više različitih nivoa privilegija (engl. *privilege level*).

Nivoi privilegija procesora

Kako bi se fizički sprečilo da korisnički programi direktno pristupaju resursima koji im nisu namenjeni, procesor obično podržava više različitih nivoa privilegija (engl. *privilege level*).

U svakom trenutku, procesor se nalazi na jednom od mogućih nivoa privilegija.

Nivoi privilegija procesora

Kako bi se fizički sprečilo da korisnički programi direktno pristupaju resursima koji im nisu namenjeni, procesor obično podržava više različitih nivoa privilegija (engl. *privilege level*).

U svakom trenutku, procesor se nalazi na jednom od mogućih nivoa privilegija.

Svaki od nivoa podrazumeva određeni skup registara i instrukcija procesora koje se mogu koristiti na tom nivou.

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.
- Nivo 0 je najviši nivo privilegija i u njemu su dostupne sve instrukcije i svi registri procesora.

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.
- Nivo 0 je najviši nivo privilegija i u njemu su dostupne sve instrukcije i svi registri procesora.
- Nivo 3 – nije moguće direktno pristupiti ni hardveru, ni registrima namenjenim za rad operativnog sistema.

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.
- Nivo 0 je najviši nivo privilegija i u njemu su dostupne sve instrukcije i svi registri procesora.
- Nivo 3 – nije moguće direktno pristupiti ni hardveru, ni registrima namenjenim za rad operativnog sistema.
- Po uključivanju računara, procesor se podrazumevano nalazi u najvišem nivou privilegija (nivo 0).

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.
- Nivo 0 je najviši nivo privilegija i u njemu su dostupne sve instrukcije i svi registri procesora.
- Nivo 3 – nije moguće direktno pristupiti ni hardveru, ni registrima namenjenim za rad operativnog sistema.
- Po uključivanju računara, procesor se podrazumevano nalazi u najvišem nivou privilegija (nivo 0).
- Posebnom instrukcijom procesor se može prebaciti u neki od nižih nivoa po želji.

Nivoi privilegija procesora

- Na arhitekturi *x86* (i *x86-64*) postoje četiri nivoa privilegija.
- Nivo 0 je najviši nivo privilegija i u njemu su dostupne sve instrukcije i svi registri procesora.
- Nivo 3 – nije moguće direktno pristupiti ni hardveru, ni registrima namenjenim za rad operativnog sistema.
- Po uključivanju računara, procesor se podrazumevano nalazi u najvišem nivou privilegija (nivo 0).
- Posebnom instrukcijom procesor se može prebaciti u neki od nižih nivoa po želji.
- Ne postoji instrukcija koja bi nam omogućila da se vratimo na viši nivo.

Procesor obično obezbeđuje jesu specijalni mehanizmi privremenog i kontrolisanog prelaska na viši nivo privilegija radi obavljanja sistemskih poslova

Operativni sistem

Podrška za upravljanje memorijom

Podrška za upravljanje memorijom

U okviru savremenih procesora postoji memorijska jedinica koja podržava mehanizam straničenja.

Podrška za upravljanje memorijom

U okviru savremenih procesora postoji memorijska jedinica koja podržava mehanizam straničenja.

Mnogi procesori podržavaju više različitih varijanti straničenja.

Podrška za upravljanje memorijom

U okviru savremenih procesora postoji memorijska jedinica koja podržava mehanizam straničenja.

Mnogi procesori podržavaju više različitih varijanti straničenja.

Na primer, x86 arhitektura podržava dve različite veličine stranica: 4KB i 4MB.

Operativni sistem

Sistem prekida

Sistem prekida

–

Sistem prekida

Predstavlja mehanizam koji omogućava zaustavljanje programa bez njegove volje u proizvoljnom trenutku i prenošenje kontrole operativnom sistemu.

Sistem prekida

–

Sistem prekida

Predstavlja mehanizam koji omogućava zaustavljanje programa bez njegove volje u proizvoljnom trenutku i prenošenje kontrole operativnom sistemu.

Bez sistema prekida ne bi bilo moguće zaustaviti program kada on jednom krene da se izvršava, osim da isključimo računar.

Sistem prekida – obradu asinhronih događaja

Asinhron događaj – izazvan interakcijom sa ulazno-izlaznim uređajima računara.

Na pimer: pomeranje miša ili pritiskanje tastature i sično.

Sistem prekida – obradu asinhronih događaja

Asinhron događaj – izazvan interakcijom sa ulazno-izlaznim uređajima računara.

Na pimer: pomeranje miša ili pritiskanje tastature i sično.

Ovakav događaj je potrebno obraditi u što kraćem roku od odgovarajućeg drajvera. Dva razloga: korisnik i memorija uređaja.

Sistem prekida – obradu asinhronih događaja

- Ulazno-izlazni uređaji aktiviraju signal prekida.

Sistem prekida – obradu asinhronih događaja

- Ulazno-izlazni uređaji aktiviraju signal prekida.
- Procesor nakon svake instrukcije proverava da li je ovaj signal aktiviran i, ako jeste, automatski zaustavlja dalje izvršavanje tekućeg procesa.

Sistem prekida – obradu asinhronih događaja

- Ulazno-izlazni uređaji aktiviraju signal prekida.
- Procesor nakon svake instrukcije proverava da li je ovaj signal aktiviran i, ako jeste, automatski zaustavlja dalje izvršavanje tekućeg procesa.
- Procesor pamti stanje registara u kontrolnom bloku procesa i zatim poziva unapred datu proceduru operativnog sistema koja je zadužena za obradu prekida.

Sistem prekida – obradu asinhronih događaja

- Ulazno-izlazni uređaji aktiviraju signal prekida.
- Procesor nakon svake instrukcije proverava da li je ovaj signal aktiviran i, ako jeste, automatski zaustavlja dalje izvršavanje tekućeg procesa.
- Procesor pamti stanje registara u kontrolnom bloku procesa i zatim poziva unapred datu proceduru operativnog sistema koja je zadužena za obradu prekida.
- Prilikom izvršavanja ove procedure, procesor obično prelazi u privilegovani režim rada.

Sistem prekida – obradu sinhronih događaja

- Drugi slučaj prekida – prilikom rada algoritma raspoređivanja (kada procesu istekne vremenski kvantum).

Sistem prekida – obradu sinhronih događaja

- Drugi slučaj prekida – prilikom rada algoritma raspoređivanja (kada procesu istekne vremenski kvantum).

- Ovaj uređaj se može posebnim komandama programirati tako da generiše periodični signal za prekid u ravnomernim intervalima.

Sistem prekida – obradu sinhronih događaja

- Drugi slučaj prekida – prilikom rada algoritma raspoređivanja (kada procesu istekne vremenski kvantum).

Obično u računarskom sistemu postoji poseban kontroler koji se naziva programibilni tajmer (engl. *programmable interval timer* (PIT)).

- Ovaj uređaj se može posebnim komandama programirati tako da generiše periodični signal za prekid u ravnomernim intervalima.

Sistem prekida – obradu sinhronih događaja

- Drugi slučaj prekida – prilikom rada algoritma raspoređivanja (kada procesu istekne vremenski kvantum).

Obično u računarskom sistemu postoji poseban kontroler koji se naziva programibilni tajmer (engl. *programmable interval timer* (PIT)).

- Ovaj uređaj se može posebnim komandama programirati tako da generiše periodični signal za prekid u ravnomernim intervalima.
- Operativni sistem obrađuje ovaj prekid tako što pokreće algoritam raspoređivanja i određuje sledeći proces koji treba da nastavi sa izvršavanjem.

Sistem prekida – mehanizam implementacije sistemskih poziva

- Sistemskim pozivima zahteva od jezgra sistema da za potrebe i u ime pozivajućeg procesa obavi neku sistemsku funkciju.

Sistem prekida – mehanizam implementacije sistemskih poziva

- Sistemskim pozivima zahteva od jezgra sistema da za potrebe i u ime pozivajućeg procesa obavi neku sistemsku funkciju.
- Ove funkcionalnosti mora obavljati jezgro sistema i mora ga obavljati u privilegovanom režimu rada.

Sistem prekida – mehanizam implementacije sistemskih poziva

- Sistemskim pozivima zahteva od jezgra sistema da za potrebe i u ime pozivajućeg procesa obavi neku sistemsku funkciju.
- Ove funkcionalnosti mora obavljati jezgro sistema i mora ga obavljati u privilegovanom režimu rada.
- Korisnički proces se izvršava u neprivilegovanom režimu $\longrightarrow$

Sistem prekida – mehanizam implementacije sistemskih poziva

- Sistemskim pozivima zahteva od jezgra sistema da za potrebe i u ime pozivajućeg procesa obavi neku sistemsku funkciju.
- Ove funkcionalnosti mora obavljati jezgro sistema i mora ga obavljati u privilegovanom režimu rada.
- Korisnički proces se izvršava u neprivilegovanom režimu $\longrightarrow$
- neophodno obezbediti kontrolisani prelazak u privilegovani režim

Sistem prekida – mehanizam implementacije sistemskih poziva

- Sistemskim pozivima zahteva od jezgra sistema da za potrebe i u ime pozivajućeg procesa obavi neku sistemsku funkciju.
- Ove funkcionalnosti mora obavljati jezgro sistema i mora ga obavljati u privilegovanom režimu rada.
- Korisnički proces se izvršava u neprivilegovanom režimu $\longrightarrow$
- neophodno obezbediti kontrolisani prelazak u privilegovani režim
- Ovo se obavlja tako što se posebnom instrukcijom procesora namerno izazove prekid. Ovakvi prekidi se nazivaju **softverski prekidi**.

Sistem prekida – neuspešna instrukcija

- Četvrti slučaj predstavljaju situacije u kojima neka instrukcija procesa iz nekog razloga ne može uspešno da se izvrši.

Sistem prekida – neuspešna instrukcija

- Četvrti slučaj predstavljaju situacije u kojima neka instrukcija procesa iz nekog razloga ne može uspešno da se izvrši.
- U takvim situacijama se automatski generiše prekid, a kontrola se predaje operativnom sistemu koji u privilegovanom režimu rada pokušava da obradi nastalu grešku.

Sistem prekida – neuspešna instrukcija

- Četvrti slučaj predstavljaju situacije u kojima neka instrukcija procesa iz nekog razloga ne može uspešno da se izvrši.
- U takvim situacijama se automatski generiše prekid, a kontrola se predaje operativnom sistemu koji u privilegovanom režimu rada pokušava da obradi nastalu grešku.
- Ovakvi prekidi poznati su i pod nazivom izuzetci (engl. *exceptions*).

Operativni sistemi zasnovani na UNIX-u

Operativni sistemi UNIX-u

- Nastao je u Belovim laboratorijama davne 1969. godine.

Operativni sistemi UNIX-u

- Nastao je u Belovim laboratorijama davne 1969. godine.
- Autor: Ken Tompson

Operativni sistemi UNIX-u

- Nastao je u Belovim laboratorijama davne 1969. godine.
- Autor: Ken Tompson
- Prve verzije UNIX-a programirane su na asemblerskom jeziku računara PDP-11.

Operativni sistemi UNIX-u

- Nastao je u Belovim laboratorijama davne 1969. godine.
- Autor: Ken Tompson
- Prve verzije UNIX-a programirane su na asemblerskom jeziku računara PDP-11.
- Denis Riči razvio je programski jezik C (pogodan za sistemsko programiranje).

Operativni sistemi UNIX-u

- Nastao je u Belovim laboratorijama davne 1969. godine.
- Autor: Ken Tompson
- Prve verzije UNIX-a programirane su na asemblerskom jeziku računara PDP-11.
- Denis Riči razvio je programski jezik C (pogodan za sistemsko programiranje).
- UNIX je ponovo napisan na C-u, mogao je da se prevodi i izvršava na različitim računarima.

GNU

- Slobodna licenca – slobodno se širio i različiti autori su doprinosili njegovom razvoju.

GNU

- Slobodna licenca – slobodno se širio i različiti autori su doprinosili njegovom razvoju.
- Početkom 80tih godina XX veka UNIX licence postaju restriktivnije. I dolazi do razvoja drugih sistema koji takođe imaju ograničene licence.

GNU

- Slobodna licenca – slobodno se širio i različiti autori su doprinosili njegovom razvoju.
- Početkom 80tih godina XX veka UNIX licence postaju restriktivnije. I dolazi do razvoja drugih sistema koji takođe imaju ograničene licence.

GNU

Kao protivteža ovakvom scenariju nastaje projekat *GNU* (engl. *GNU is Not Unix* (GNU)) sa ciljem razvoja operativnog sistema kompatibilnog sa UNIX-om koji će garantovano zauvek biti distribuiran pod slobodnom licencom.

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).
- Kao i kompletno korisničko okruženje operativnog sistema.

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).
- Kao i kompletno korisničko okruženje operativnog sistema.
- Ono što je nedostajalo bio je najkomplikovaniji deo operativnog sistema – jezgro.

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).
- Kao i kompletno korisničko okruženje operativnog sistema.
- Ono što je nedostajalo bio je najkomplikovaniji deo operativnog sistema – jezgro.
- *Linux* jezgro – početkom 90tih godina razvio finski programer Linus Torvalds.

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).
- Kao i kompletno korisničko okruženje operativnog sistema.
- Ono što je nedostajalo bio je najkomplikovaniji deo operativnog sistema – jezgro.
- *Linux* jezgro – početkom 90tih godina razvio finski programer Linus Torvalds.
- Ugradnjom Linux jezgra u GNU okruženje nastaje *GNU/Linux* operativni sistem.

GNU/Linux

- Najpre razvijeni programerski alati(editor, prevodilac, debager, linker, i sl.).
- Kao i kompletno korisničko okruženje operativnog sistema.
- Ono što je nedostajalo bio je najkomplikovaniji deo operativnog sistema – jezgro.
- *Linux* jezgro – početkom 90tih godina razvio finski programer Linus Torvalds.
- Ugradnjom Linux jezgra u GNU okruženje nastaje *GNU/Linux* operativni sistem.
- Različite distribucije: *Ubuntu*, *Mint*, *Debian*, *Fedora*, *Slackware* i sl.

Komandno okruženje operativnog sistema UNIX

- Tradicionalni korisnički interfejs za UNIX-zasnovane operativne sisteme je školjka (engl. *shell*).

- Tradicionalni korisnički interfejs za UNIX-zasnovane operativne sisteme je školjka (engl. *shell*).
- U pitanju je tekstualni korisnički interfejs koji omogućava jednostavno pokretanje komandi sa zadavanjem opcija komandne linije.

- Tradicionalni korisnički interfejs za UNIX-zasnovane operativne sisteme je školjka (engl. *shell*).
- U pitanju je tekstualni korisnički interfejs koji omogućava jednostavno pokretanje komandi sa zadavanjem opcija komandne linije.
- Pre pojave grafičkih korisničkih interfejsa, ljuska je bila podrazumevani korisnički interfejs na UNIX-zasnovanim sistemima.

- Tradicionalni korisnički interfejs za UNIX-zasnovane operativne sisteme je školjka (engl. *shell*).
- U pitanju je tekstualni korisnički interfejs koji omogućava jednostavno pokretanje komandi sa zadavanjem opcija komandne linije.
- Pre pojave grafičkih korisničkih interfejsa, ljuska je bila podrazumevani korisnički interfejs na UNIX-zasnovanim sistemima.
- Danas – potrebno je pokrenuti aplikaciju sa grafičkim interfejsom koja predstavlja emulator terminala (često se zove Terminal ili Console).