**2. Upravljanje procesima**

**Koncept procesa**

* Svi multiprogramski OS zasnovani su na konceptu procesa
* Operativni sistem mora da obezbedi:
  + Kreiranje procesa
  + Preplitanje izvršenja više procesa u cilju maksimalnog iskorišćenja procesora
  + Dodelu resursa procesima u skladu sa određenom politikom
  + Sinhronizaciju i uzajmno isključivanje procesa i zaštitu resursa svakog procesa od strane drugih procesa
  + Kounikaciju između procesa

**Osnovni pojmovi i koncepti**

* Računarska platforma se sastoji od skupa hardverskih resursa
* Računarske aplikacije se razvijaju kako bi se izvršile neki zadatak; prihvataju ulaz spolja, vrše obradu i generišu izlaz
* Nije efikasno pisati aplikacije direktno za datu hardversku platformu
* OS obezbeđuje interfejs između aplikacije i hardvera računara
* OS pruža apstrakciju resursa, koje aplikacije mogu tražiti ili pristupati, i upravljati njihovom upotrebom

**Šta je proces?**

* Proces je osnovni koncept operativnog sistema i predstavlja program u izvršavanju na procesoru
  + Proces se ponekad naziva i zadatak (task)
  + Svi multiprogramski OS su izgrađeni oko koncepta procesa
* Gledano sa strane OS-a, proces je osnovna jedinica izvršavanja i najmanji entit koji se može planirati, dodeliti i izvršavati na procesoru
* Proces je jedinica aktivnosti koju karakteriše izvršenje sekvence instrukcija, trenutno stanje i pridružen skup sistemskih resursa

**Elementi procesa**

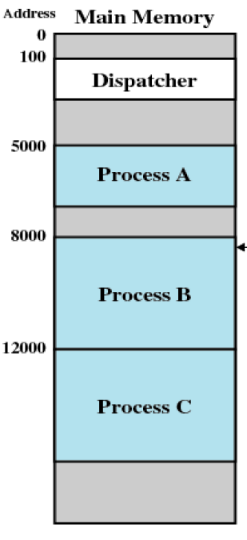
* Proces se sastoji od:
  + Programskog koda
  + Skupa podataka nad kojima se izvršavaju instrukcije
  + Steka
  + Atributa koji opisuju stanje procesa i koji su smešteni u
    - UPRAVLJAČKI BLOK PROCESA:
      * Identifikator
      * Stanje
      * Prioritet
      * Programski brojač (Program counter)
      * Pokazivač na memoriju u koju je smešten proces
      * Konktekstni podaci (podaci smešteni u registrima procesora za vreme izvršenja procesa)
      * Informacije o U/I statusu
      * Informacije o obračunu korišćenja resursa...

**Upravljački blok procesa**

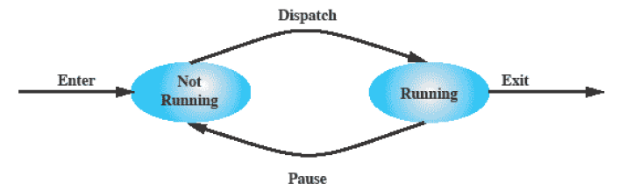
* Proces Control Block – PCB
* Sadrži atribute procesa
* Kreiran i upravljan od strane OS-a
* Obezbeđuje podršku za višestruke procese
* Tabela procesa predstavlja skup PCB-ova svih procesa

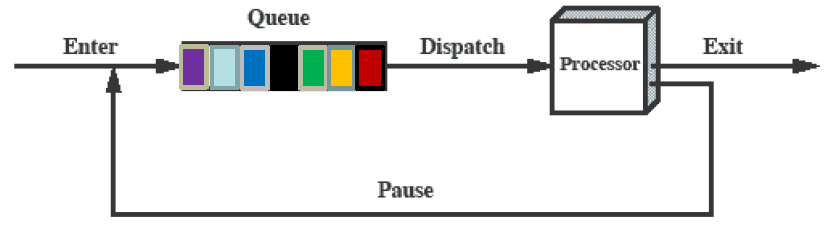
**Praćenje (tok, trace) procesa**

* Ponašanje individualnog procesa je prikazano sekvencom instrukcija koje se izvršavaju
* Ova lista instrukcija se naziva praćenje (tok, trace) procesa
* Dispečer (raspoređivač, dispatcher) je program koji prebacuje (switch) menja procese sa trenutno aktivnim procesom na procesoru (this is so stupid)

**Izvršenje procesa**

* Razmotrićemo tri procesa koji se izvršavaju
* Svi procesi su u glavnoj memoriji, uključujući i dispatcher
* Uovom primeru ćemo ignorisati postojanje virtualne memorije
* Programski brojač u CPU sadrži adresu instrukcije koja treba da bude izvršena
* Kada se aktivira sledeći proces menja se sadržaj programskog brojača i njegova vrednost postavlja na adresu instrukcije koju treba izvršiti u novom procesu
* SVAKI PROCES SE IZVRŠAVA DO SVOG ZAVRŠETKA

**Model dva stanja procesa**

* Proces može biti u jednom od dva stanja
  + Izvršava se
  + Ne izvršava se
* Red procesa
  + Procesi koji se ne izvršavaju se smeštaju u red (elementi su pokazivači na PCB), izvršavaju se podređeno vreme i vraćaju u red dok se ne završe.

**Upravljanje procesima u OS**

* Osnovne funkcije:
  + Kreiranje i završavanje procesa
  + Suspendovanje i ponovno aktiviranje
  + Planiranje izvršenja procesa i upravljanje procesorom/procesima
  + Obezbeđivanje mehanizma za sinhronizaciju i komunikaciju između procesa
  + Obezbeđenje mehanizma za upravljanje deadlock-om (uzajamno blokiranje, samrtni zagljaj, zastoj)

**Kreiranje (stvaranje) procesa**

* Glavni razlozi za kreiranje procesa:

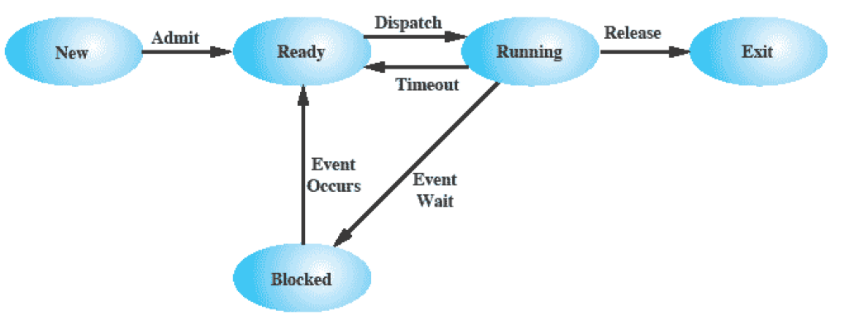
1. Iniciranje nekog paketnog (batch) posla
2. Interaktivna prijava (login) korisnika
3. Kreiran od strane OS za obezbeđenje nekog servisa
4. Kreiran od strane postojećeg procesa

* Roditeljski (parent) proces i dete (child, potomak)
* Aktivnosti OS pri kreiranju procesa:

1. Dodela identifikatora procesu
2. Dodela inicijalnog prioriteta procesu
3. Kreiranje i unos PCB novog procesa u Tabelu procesa
4. Dodela početnih resursa procesu (memorija, otvorene datoteke itd)

**Završetak (terminiranje) procesa**

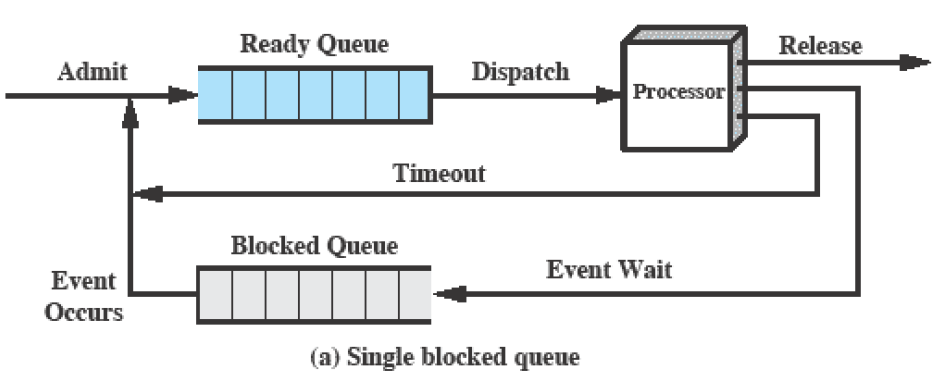
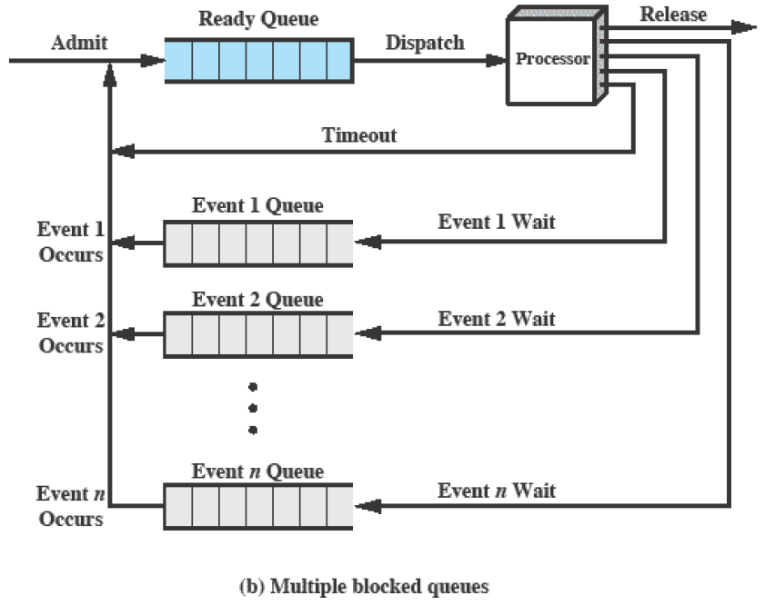
1. Normalni završetak
2. Isteklo dodeljeno vreme za izvršavanje
3. Nedovoljno raspoložive memorije
4. Greška usled narušavanja zaštite (protection error) u pristupu memoriji ili drugim resursima bez autorizacije
5. Aritmetička greška (deljenje nulom)
6. Prekoračenje vremena čekanja na neki događaj
7. U/I greška, pogrešna instrukcija, privilegovana instrukcija, pogrešna upotreba podataka, itd
8. Ubijen od OS ili oepratera
9. Završetak roditeljskog procesa
10. Na zahtev roditeljskog procesa koji ima privilegije da završi procese decu

**Model procesa sa pet stanja**

* **Izvršavanje (Running)** – Proces koji se trenutno izvršava
* **Spreman (Ready**) – Proces koji je spreman za izvršavanje kada mu se pruža prilika
* **Blokiran (Blocked)** – Proces ne može da se izvršava dok se ne desi neki događaj, poput završetka U/I operacije
* **Novi (New)** – Proces tek kreiran, ali još ubek nije primljen od strane OS u skup izvršnih procesa
* **Završen (Exit)** – Proces izbačen iz skupa izvršnih procesa od strane OS-a

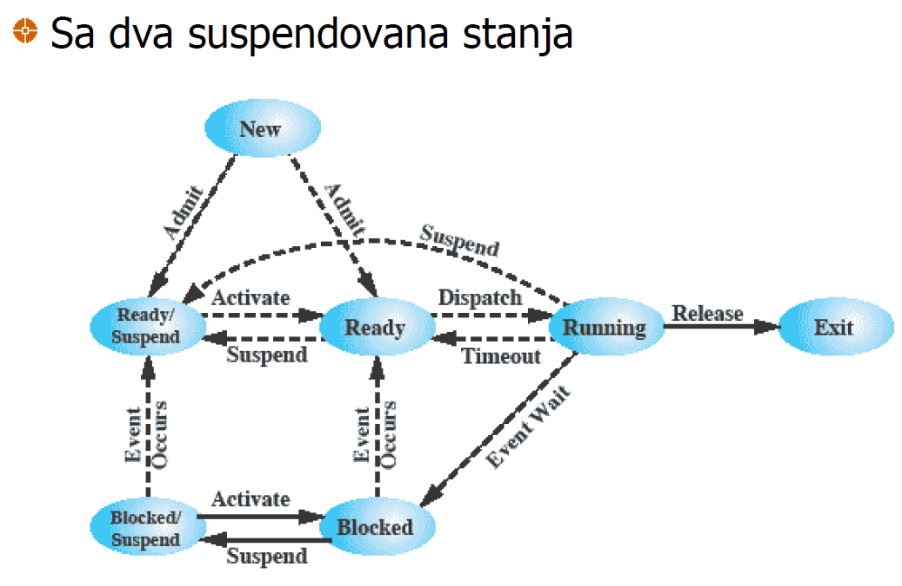
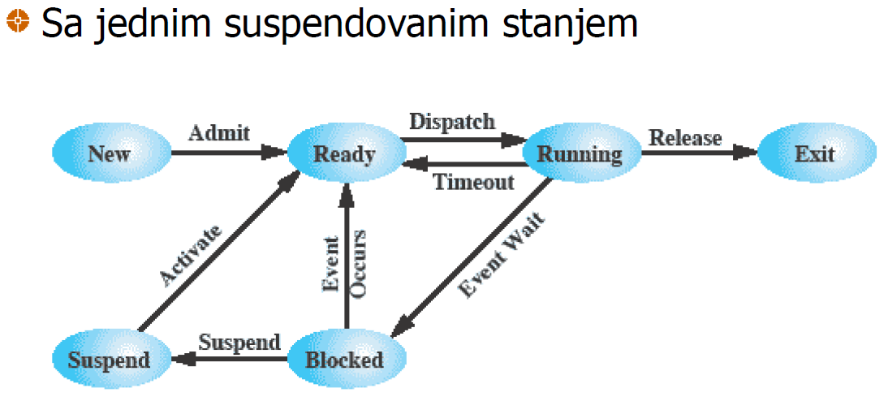
**Prelazi izeđu stanja procesa**

* **Null -> Novi**
  + Kreiranje procesa za izvršavanje programa
* **Novi -> Spreman**
  + OS prebacuje proces kada ima dovoljno resursa za njegovo aktiviranje
* **Spreman -> Izvršavanje**
  + OS bira jedan od spremnih procesa za izvršavanje na CPU
* **Izvršavanje -> Završen**
  + Trenutno aktivan proces se završava od strane OS iz nekog od razloga
* **Izvršavanje -> Spreman**
  + Isteklo je vreme dodeljeno procesu za izvršavanje, ili je proces višeg prioriteta postao spreman
* **Izvršavanje -> Blokiran**
  + Proces je zahtevao resurs zbog koga mora da čeka (datoteka, memorijska sekcija, poruka od drugog procesa) ili U/I operaciju koja mora biti završena pre nastavka procesa
* **Blokiran -> Spreman**
  + Nastao je događaj na koji je proces čekao
* **Spreman -> Završen**
  + Nije prikazano na dijagramu ; u nekim sistemima roditeljski proces može završiti proces decu u bilo kom trenutku, pa i kada su u stanju spreman, a takođe završetkom roditeljskog procesa završavaju se i sva njegova deca procesi
* **Blokiran -> Završen**

**Redovi procesa**

**Suspendovani procesi**

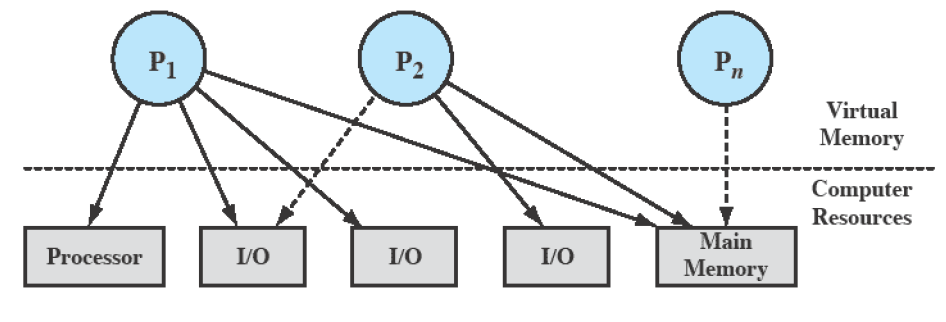
* Procesor je mnogo brži od U/I uređaja, tako da može da se desi da nogi (svi) procesi čekaju na završetak U/I operacija
  + Prebaciti (Swap) ove procese iz memorije na disk radi oslobađanja više memorije i koristiti procesor za aktiviranje novih procesa ili prethodno suspendovanih procesa
* Proces u stanju blokiran prelazi u stanje suspendovan kada se prebaci na disk
* Dva nova stanja
  + Blokiran/Suspendovan (Blocked/Suspended)
  + Spreman/Suspendovan (Ready/Suspended)

**Dijagram prelaza stanja procesa**

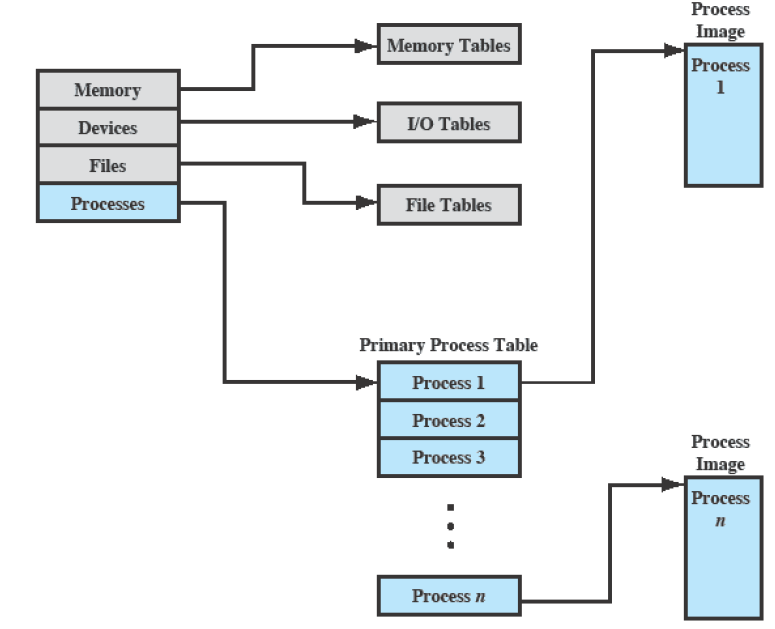
**Razlozi za suspendovanje procesa**

* **Prebacivanje (swapping)** – OS mora da oslobodi dovoljno glavne memorije da aktivira proces koji je spreman za izvršavanje
* **Drugi OS razlozi** – OS može suspendovati proces za koji se sumnja da izaziva “probleme” tokom izvršavanja
* **Zahtev interaktivnog korisnika** – Suspendovanje procesa u toku debagiranja
* **Vremenski** – Proces se izvršava periodično; Između perioda može biti suspendovan
* **Roditeljski proces** – može suspendovati izvršenje procesa deteta.

**Procesi i resursi**

* Procesi i njima dodeljeni (alocirani) resursi u nekom vremenskom trenutku
* Da bi OS upravljao procesima i resursima on mora imati informacije o trenutnom stanju svakog procesa i resursa.
* Za svaki entitet kojim upravlja, OS kreira odgovarajuće tabele

**Upravljačke tabele OS-a**

* Generalna struktura upravljačkih tabela operativnog sistema

**Tabela procesa**

* Da bi upravljao procesima OS mora da u svakom trenutku zna detalje o svakom procesu
  + Trenutno stanje
  + Identifikator procesa (ID procesa)
  + Lokaciju u memoriji
  + Ostale atribute procesa
* **Upravljački blok procesa (Procss Control Block – PCB)** – kolekcija atributa procesa (**Deskriptor procesa)**
  + Najvažnija struktura podataka u OS-u; Sadrži sve informacije o procesima neophodne OS-u – skup PCB-a definiše stanje OS
* **Slika procesa (Process Image)** – predstavlja skup koji čine program, podaci, magacin (stek, stack) i atributi procesa (PCB) smešteni u glavnu memoriju (delom i na disku)

**Atributi procesa**

* Informacije u okviru upravljačkog bloka procesa (PCB) mogu se grupisati u tri kategorije:
  + **Identifikatori procesa**
  + **Informacije o stanju procesora**
  + **Informacije za upravljanje procesom**

**Identifikatori procesa**

* Svakom procesu je dodeljen jedinstveni numerički identifikator (Proces ID, PID)
* Svaki proces može da poseduje identifikatore roditeljskog procesa, grupe procesa korisnika koji je kreirao proces, itd.
* Sve druge tabele kojima upravlja OS mogu da koriste identifikator procesa da bi cross-reference PCD procesa

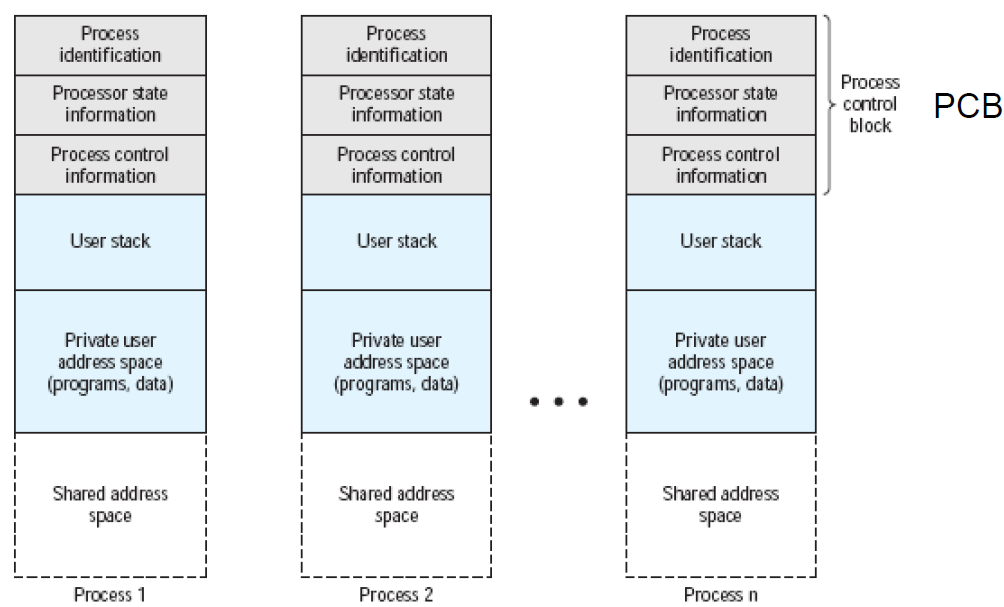
**Informacije o stanju procesora**

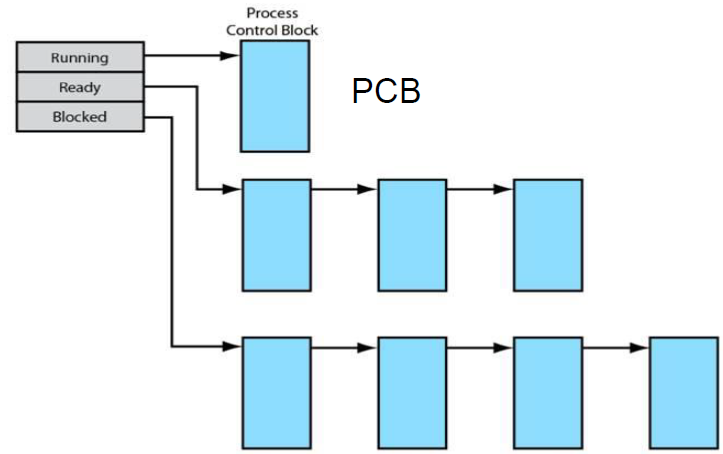
* Sastoje se od sadržaja registara procesora.
  + Registara opšte namene vidljivih od strane korsinika
  + Upravljačkih i statusnih registara (PC, uslovni kodovi, flagovi za dozvoljen/nedozvoljen prekid, mod izvršenja,...)
  + Stack pointer-a – LIFO strukture za smeštanje parametara, povratnih adresa i lokalnih promenljivih pri pozivima procedura i sistemskim pozivima
* Program Status Word (PSW)
  + Sadrži statusne informacije
  + Primer
    - EFLAGS registar na PEntium procesorima
    - Italinum (IA-64) – psr registar ima polje od dva bita: cpl (0 najviše privilegija, 3 najmanje privilegija)

**Informacije za upravljanje procesom**

* Informacije o stanju i planiranju (scheduling)
  + Stanje procesa
  + Prioritet
  + Informacije vezane za planiranje – zavisno od algoritma planiranja
  + Događaj na koji proces čeka
* Struktuiranje (povezivanje, ulančavanje) PCB-a
  + Procesi (PCB) su povezani u različitim strukturama (redovima, listama, itd): roditelj – dete, red čekanja na U/I, itd.
* Međuprocesorska komunikacije
  + Različiti flag-ovi, signali, poruke između procesa
* Privilegije procesa
  + Privilegije za pristup memoriji, izvršavanje određenih tipova instrukcija, sistemskih servisa
* Upravljanje memorijom
  + Pokazivači (adrese) na tabele stranica/segmenata koje opisuju virtuelnu memoriju dodeljenu procesu
* Vlasništvo i korišćenje resursa
  + Resursi kojima upravlja proces, poput otvorenih datoteka
  + Takođe, moeže biti uključena i istorija korišćenja procesora ili ostalih resursa za potrebe planiranja procesa

**Slika procesa u virtuelnoj memoriji**

* Struktura slike procesa u virtuelnoj memoriji

**Struktura redova (listi) procesa**

* PCB je najvažnija struktura podataka u operativnom sistemu (Deskriptor procesa)
* OS moduli (planiranje, alokacija resursa, obrada prekida, nadgledanje i analiza performansi, itd) čitaju i/ili modifikuju PCB-ove
* PCB su povezani u različitim redovima i tabelama koji se često implementiraju kao lančane liste

**Upravljanje izvršenjem procesa**

* Većina procesora podržava najmanje dva moda (režima) izvršavanja:
  + **KORISNIČKI (user) mode (režim)**
    - Režim izvršavanja sa manje privilegija – zabranjen pristup svim memorijskim adresama i izvršavanje privilegovanih (U/I) instrukcija
    - Korisnički programi se izvršavaju u ovom režimu (modu)
  + **KERNEL (system, supervisor) mode (režim) – mode jezgra**
    - Privilegovan režim izvršavanja
    - U ovom režimu (mode-u) se izvršava kernel operativnog sistema
    - Bit-ovi u PSW označavaju trenutni režim izvršenja
* Intel Itanium procesor IA-64 (IS THIS SHIT REALLY SOMETHING I’LL REMEMBER?)

**Kreiranje procesa**

* Prilikom kreiranja novog procesa OS treba da:
  + Dodeli jedinstveni identifikator procesa
  + Dodeli (alocira) memorijski prostor za proces, tačnije za sve elemente slike procesa
  + Inicijalizuje upravljački blok procesa (PCB)
  + Uspostavi odgovarajuće veze novog porcesa ulančavanjem njegovog PCB-a u odgovarajuće redove/liste
  + Kreira ili proširi ostale strukture podataka
    - Na primer, za obračun korišćenja resursa, ili ocenu ili analizu performansi

**Zamena (komutiranje) procesa**

* Zamena (switch) procesa predstavlja zaenu aktivnog procesa jednim od spremnih procesa
* Događaji koji zahtevaju izvršenje koda OS-a i mogu uzrokovati zamenu procesa:
  + **Prekid (interrupt)** – eksterni događaj u odnosu na izvršavanje procesa; Prekid se javlja kao reakcija na eksterne, asinhrone događaje
    - Prekid generatora takta, U/I prekid, greška u referenci memorije
  + **Trap –** Odnosi se na izvršavanje tekuće instrukcije procesa – ako nastane greška ili izuzetak koji je fatalan
  + **Sistemski poziv (supervisor call) –** Zahtev od strane procesa koji se izvršava za izvršavanje U/I operacije (npr. pristup datoteci) kojim se aktivira odgovarajuća rutaina OS-a.

**Promena (komutiranje) režima**

* Nakon izvršenja svake isntrukcije, procesor proverava da li je nastao prekid (interrupt signal)
* Ukoliko je prekid nastao, porcesor izvršava sledeće korake:
  + Postavlja programski brojač na početnu adresu rutine za obradu perkida (interrupt handler)
  + Prebacuje se iz korisničkog u kernel režim, tako da rutina za obradu prekida može uključiti i privilegovane instrukcije
  + Kontekst prekinutog procesa se čuva u njegovom PCB-u
    - Informacije o stanju procesa
  + Izvršava se rutina za obraadu prekida (interrupt handler)
  + Ukoliko nastanak prekida ne zahteva zamenu (komutiranje) procesa i promenu stanja procesa, obnavljaju se informacije o stanju procesa (kontekst) prekinutog procesora
    - U suprotnom se vrši komutiranje porcesa i promena njegovog stanja
  + Prekid kloka ili poziv U/I operacije uzrokuje prebacivanje prekinutog procesa u novo stanje (Spreman, Blokiran)

**Promena stanja procesa**

* Zamena procesa uzrokuje promenu njegovog stanja
* Koraci koje obavlja OS pri zameni procesa:

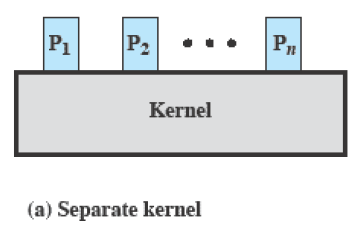
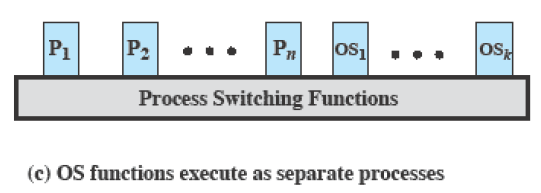
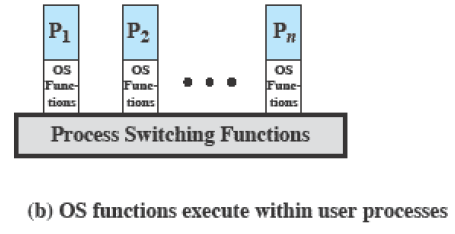
1. Snimiti sadržaj registara procesora, programski brojač i statusni registar procesora u PCB procesa
2. Ažurirati PCB procesa koji je do tada bio u stanju IZVRŠAVANJE (RUNNING)
3. Premestiti PCB u odgovarajući red – SPREMAN, BLOKIRAN, SPREMAN/BLOKIRAN i promeniti mu stanje
4. Izabrati sledeći proces za izvršavanje
5. Ažurirati PCB procesa koji je izabran u stanje IZVRŠAVANJE
6. Ažurirati strukture podataka za upravljanje memorijom
7. Obnoviti sadržaj registara procesora na vrednosti koje su postojale u trenutku kad je proces promenjen iz stanja IZVRŠAVANJE

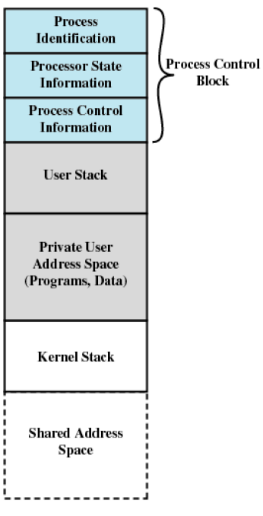
**Izvršavanje operativnog sistema**

* OS radi na isti način kao i ostali računarski softver
* Ako je OS samo kolekcija programa i ako se ovi programi izvršavaju od strane procesora baš kao i bilo koji drugi programi, da li je OS proces?
* Ako jeste, kako se njime upravlja?
  + Ko (šta) upravlja njime?

**Izvršavanje OS-a**

* **Kernel bez procesa**
  + Tradicionalni pristup u starijim OS-ima
  + Izvršavanje kernela van bilo kog procesa
  + OS kod se izvršava kao poseban entitet u glavnoj memoriji, sa svojim stekom, u kernel režimu (privilegovanom režimu)
* **Izvršavanje OS unutar korisničkih procesa**
  + Uobičajeno u OS-a na PC-u i radnim stanicama
  + OS softver predstavlja kolekciju procedura koje korisnički proces može pozvati za izvršenje različitih funkcija i koje se izvršavaju unutar okruženja korisničkog procesa
  + Slika procesa sadrži i delove za porgram, podatke i stek kernel programa
  + Proces se izvrava u kernel režimu kada se izvršava kod OS-a
* **OS zasnovan na procesima**
  + OS implemtiran kao kolekcija sistemskih procesa
  + Korisno je u višeprogramskom ili višeprogramskom okruženju

**Odnos između OS-a i korisničkog procesa**

**Slika procesa kada se OS izvršava unutar korisničkog prostora**

* OS se izvršava unutar korisničkog procesa (adresnog procesa)
* Kernel stek se koristi za upravljanje pozivima procedura dok je proces u kernel režimu
* OS kod i podaci su u deljenom adresnom prostoru (deljeni su između svih korisničkih procesa)