Operativni sistemi: interna struktura i principi dizajna

Poglavlje 3
Opis i kontrola procesa

Sedmo izdanje

William Stallings

Kako OS upravlja izvršavanjem aplikacija

- Resursi se stavljaju na raspolaganje većem broju aplikacija
- Procesor se prebacuje između više aplikacija, tako da izgleda da sve napreduju
- Procesor i U/I uređaji mogu se efikasno koristiti

Elementi procesa

- Dva osnovna elementa procesa su:

Programski kod

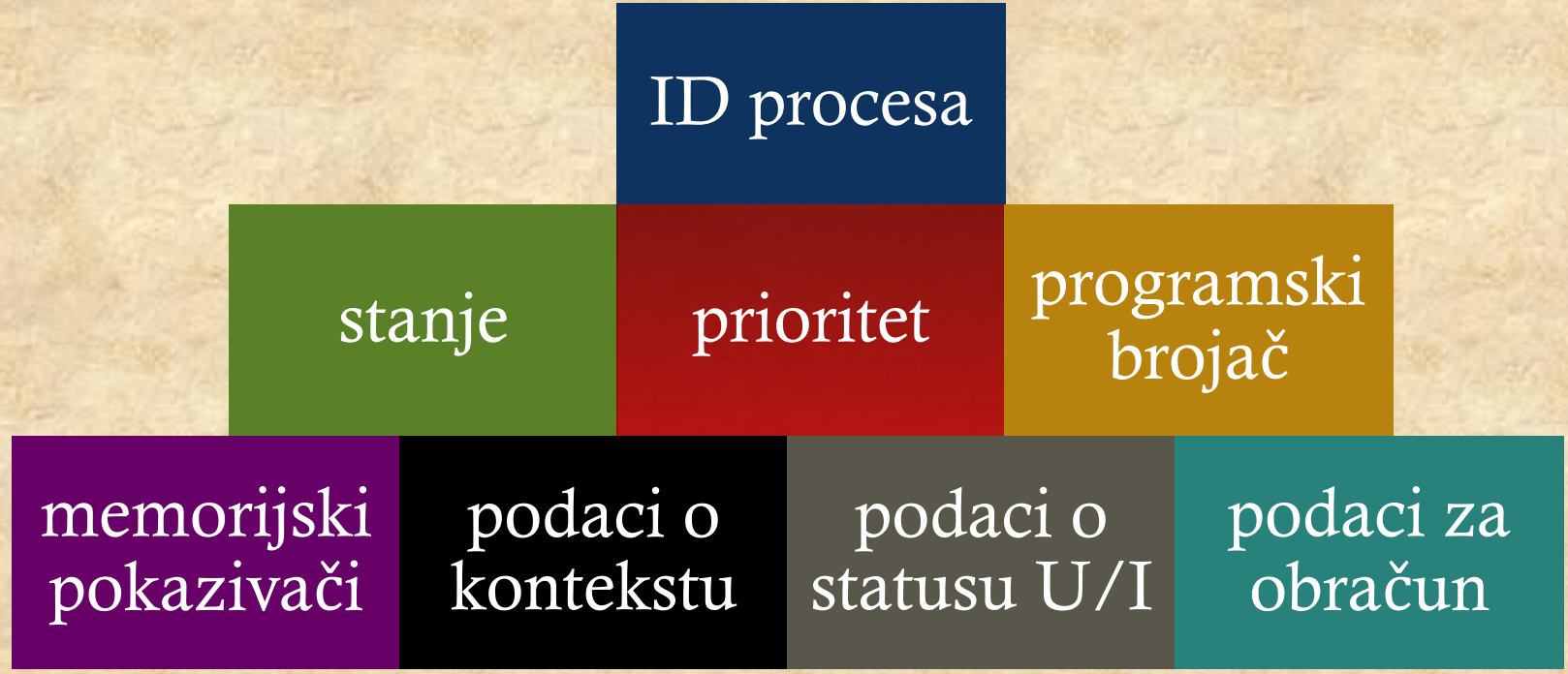
  - koji mogu deliti i drugi procesi koji izvršavaju isti program

Skup podataka pridruženih tom kodu

- Kada procesor počne da izvršava programski kod, taj entitet u izvršavanju nazivamo ***procesom***

Elementi procesa

- Dok se program izvršava, proces se može jednoznačno opisati nizom elemenata, među kojima su:



Procesni kontrolni blok

- Sadrži elemente procesa
- Omogućava da se proces koji se izvršava prekine i kasnije nastavi, kao da prekida nije ni bilo
- Kreira ga i njime upravlja operativni sistem
- Ključni alat koji omogućava podršku za više procesa (multiprogramiranje)

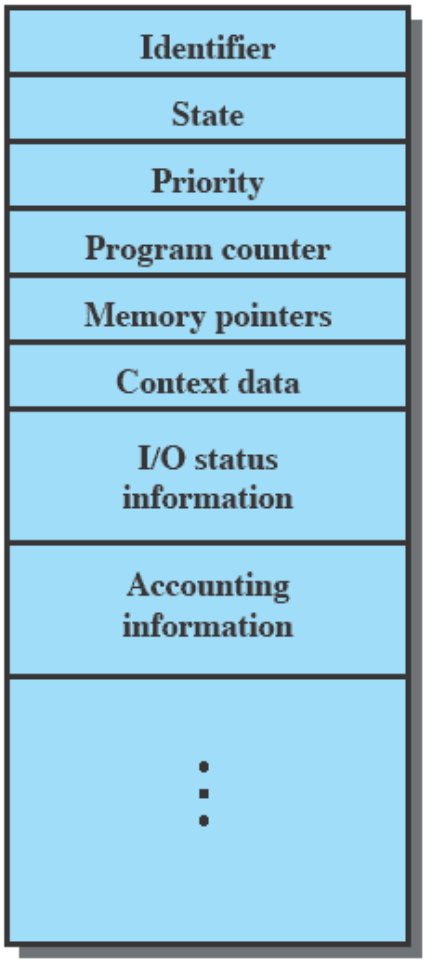
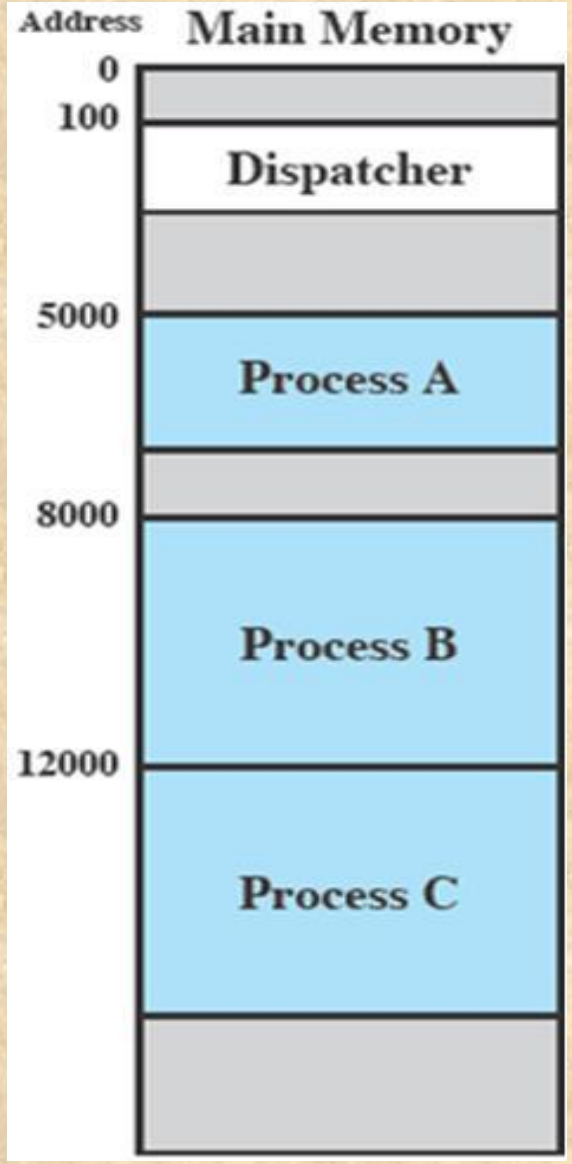


Figure 3.1 Simplified Process Control Block

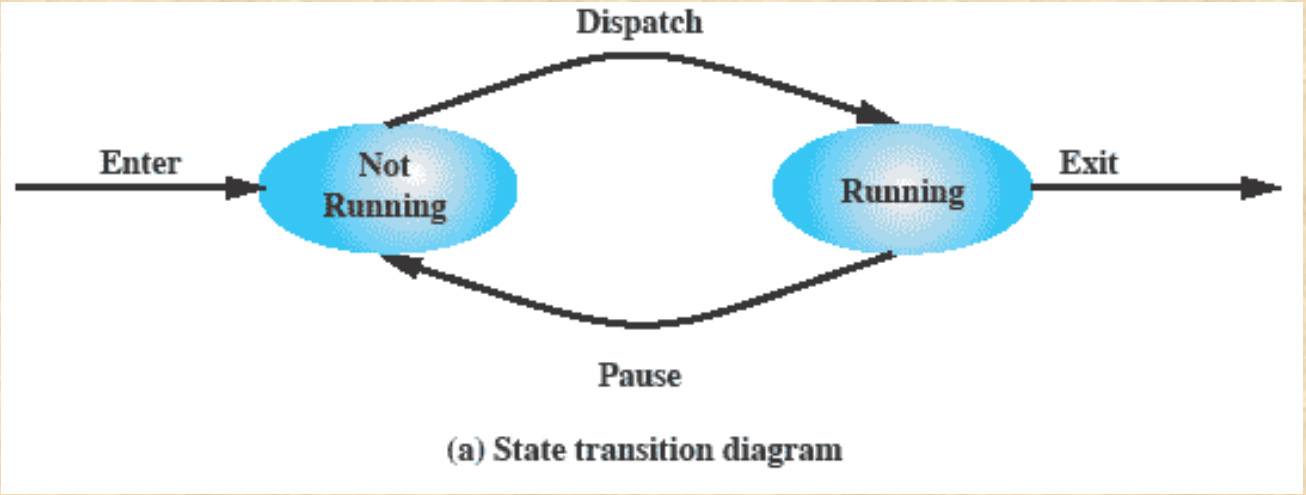
Izvršavanje procesa





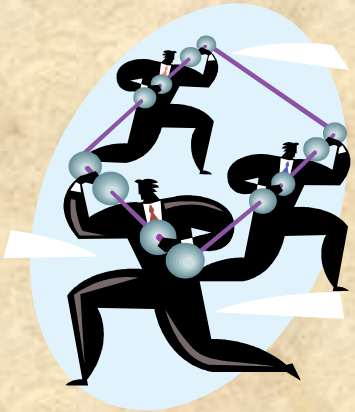
Model sa dva stanja

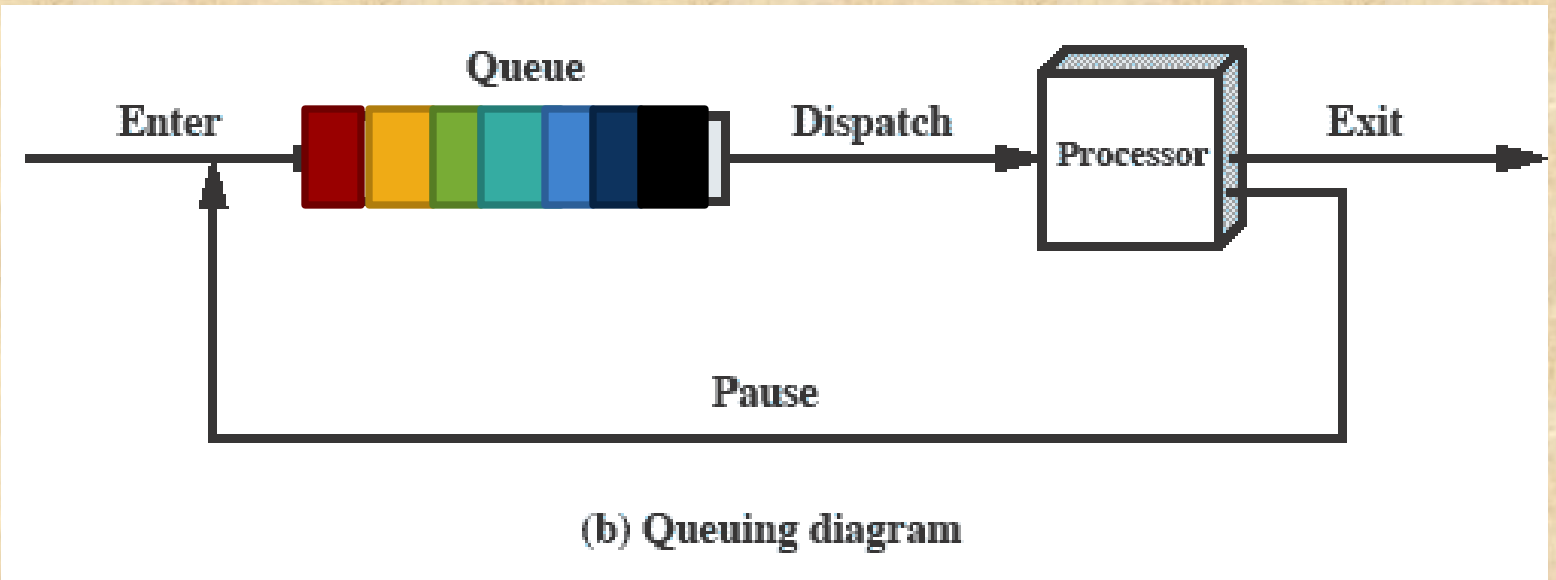
- Proces može biti u jednom od dva stanja:
  - izvršava se
  - ne izvršava se



(a) State transition diagram

Dijagram redova čekanja





(b) Queuing diagram

Tabela 3.1 Razlozi za kreiranje procesa

Novi paketni posao	OS dobija tok komandi za upravljanje paketnim poslovima, obično na traci ili disku. Kada je spreman da preuzme novi posao, OS čita sledeći niz komandi za upravljanje poslovima.
Interaktivna prijava	Korisnik se sa terminala prijavljuje na sistem.
Kreira ga OS radi pružanja usluge	OS može kreirati proces koji obavlja neku funkciju u ime korisničkog programa, a da korisnik ne mora da čeka (npr. proces koji upravlja štampanjem).
Stvara ga postojeći proces	Radi modularnosti ili iskorišćavanja paralelizma, korisnički program može zahtevati kreiranje više procesa.

Kreiranje procesa

Stvaranje procesa	*Roditeljski proces*	*Proces dete*
• kada OS kreira proces na izričit zahtev drugog procesa	• je izvorni proces, koji kreira novi	• je novi proces

Završetak procesa

- Mora postojati način da proces signalizira svoj završetak
- Paketni posao treba da sadrži instrukciju HALT ili eksplicitan poziv servisa OS-a za završetak
- Kod interaktivne aplikacije, akcija korisnika pokazuje kada je proces završen (npr. odjavljivanje, izlazak iz aplikacije)



Tabela 3.2

Razlozi za završetak procesa

Normalan završetak	Proces izvršava poziv servisa OS-a kojim signalizira da je završio rad.
Prekoračen vremenski limit	Proces se izvršavao duže od zadatog ukupnog vremenskog ograničenja. Postoji više mogućnosti za vrstu vremena koje se meri: ukupno proteklo vreme („vreme po zidnom satu“), vreme provedeno u izvršavanju i, kod interaktivnog procesa, vreme proteklo od poslednjeg unosa korisnika.
Nedovoljno memorije	Proces zahteva više memorije nego što sistem može da obezbedi.
Prekoračenje granica	Proces pokušava da pristupi memorijskoj lokaciji kojoj nema pravo da pristupa.
Greška zaštite	Proces pokušava da koristi resurs, npr. datoteku, koji nema pravo da koristi, ili pokušava da ga koristi na nedozvoljen način, npr. da upisuje u datoteku koja je samo za čitanje.
Aritmetička greška	Proces pokušava nedozvoljeno izračunavanje, npr. deljenje nulom, ili pokušava da smesti brojeve veće od onih koje hardver podržava.
Prekoračenje vremena čekanja	Proces je na neki događaj čekao duže od zadatog maksimuma.
U/I greška	Došlo je do greške pri ulazu ili izlazu, kao što je nemogućnost pronalaženja datoteke, neuspešno čitanje ili upis posle zadatog maksimalnog broja pokušaja (kada se, na primer, naiđe na oštećen deo trake) ili neispravna operacija (npr. čitanje sa linijskog štampača).
Neispravna instrukcija	Proces pokušava da izvrši nepostojeću instrukciju (često zbog skoka u oblast podataka i pokušaja da se podaci izvrše).
Privilegovana instrukcija	Proces pokušava da koristi instrukciju rezervisanu za operativni sistem.
Pogrešna upotreba podataka	Podatak je pogrešnog tipa ili nije inicijalizovan.
Intervencija operatera ili OS-a	Operater ili operativni sistem je iz nekog razloga okončao proces (npr. ako postoji mrtva petlja).
Završetak roditelja	Kada se roditeljski proces završi, operativni sistem može automatski da okonča sve njegove potomke.
Zahtev roditelja	Roditeljski proces obično ima ovlašćenje da okonča bilo kog svog potomka.



Model sa pet stanja

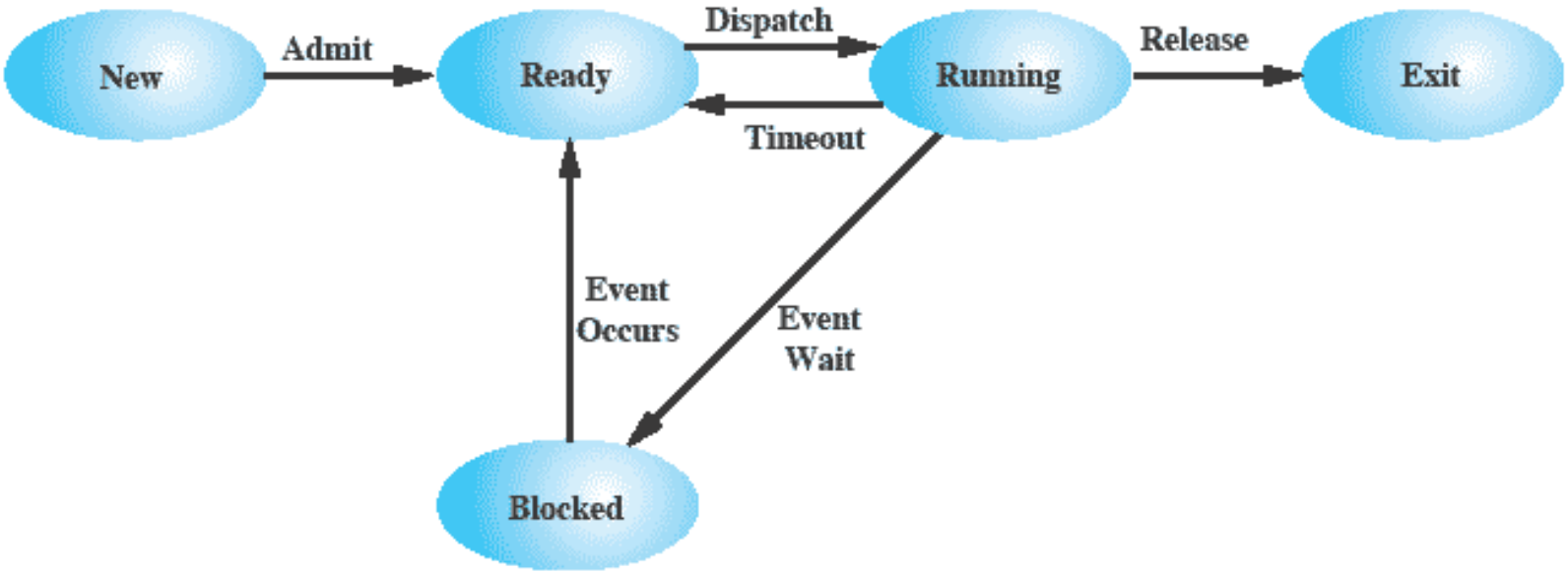


Figure 3.6 Five-State Process Model

Primer stanja procesa

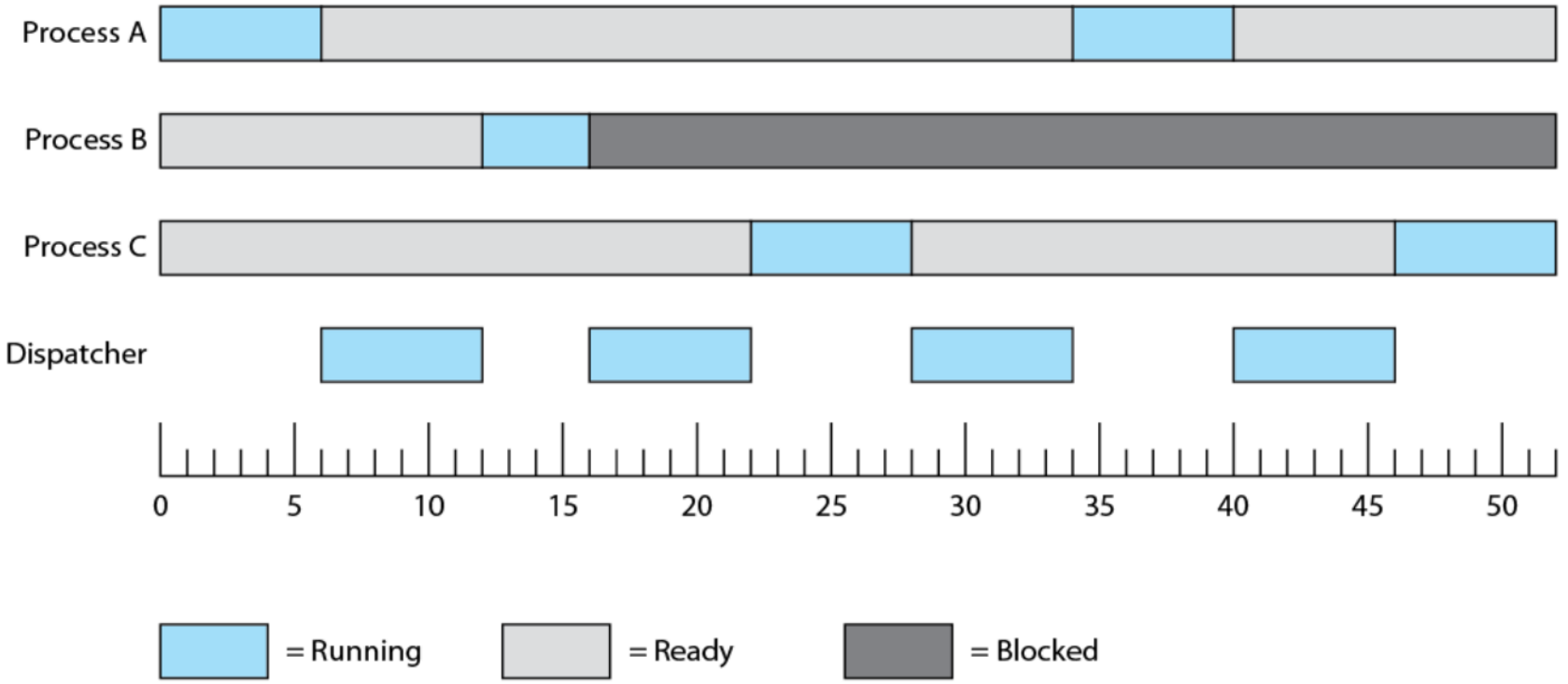
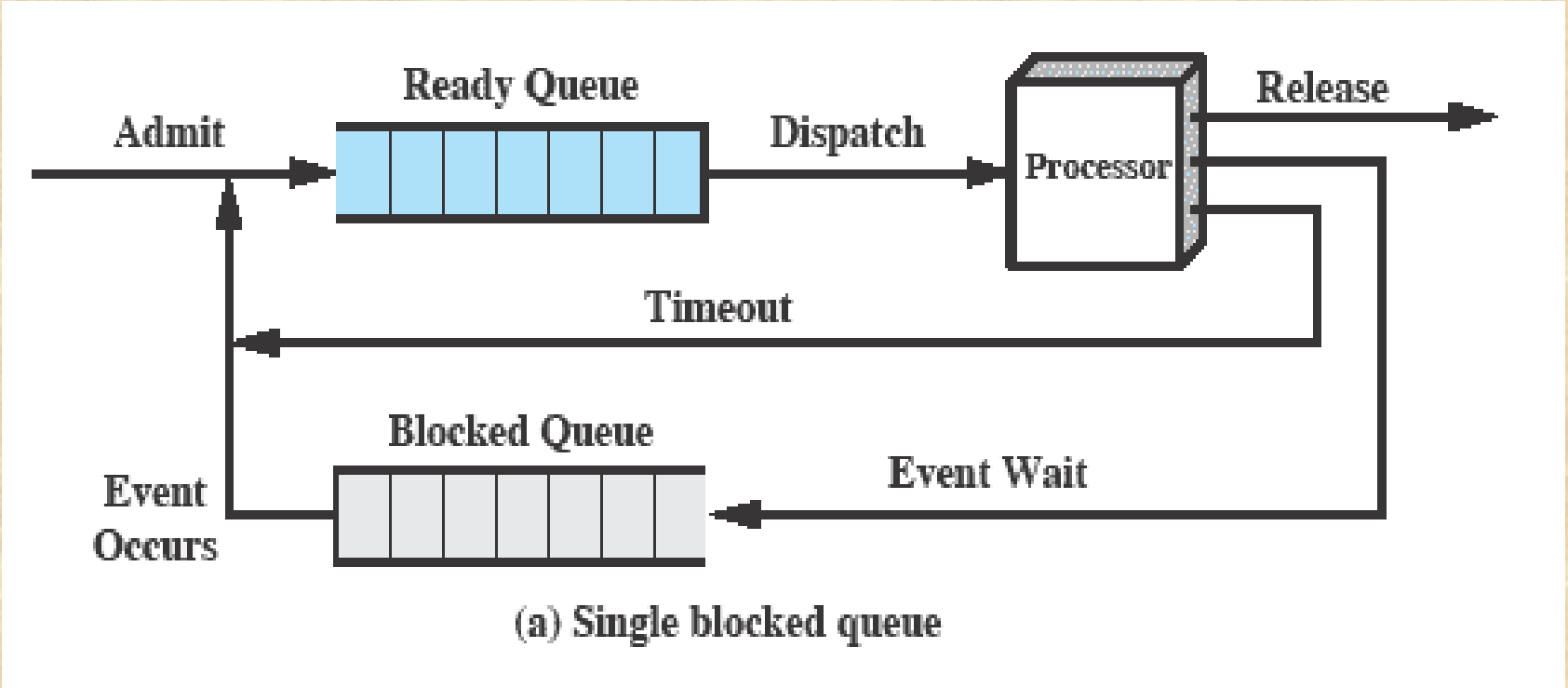


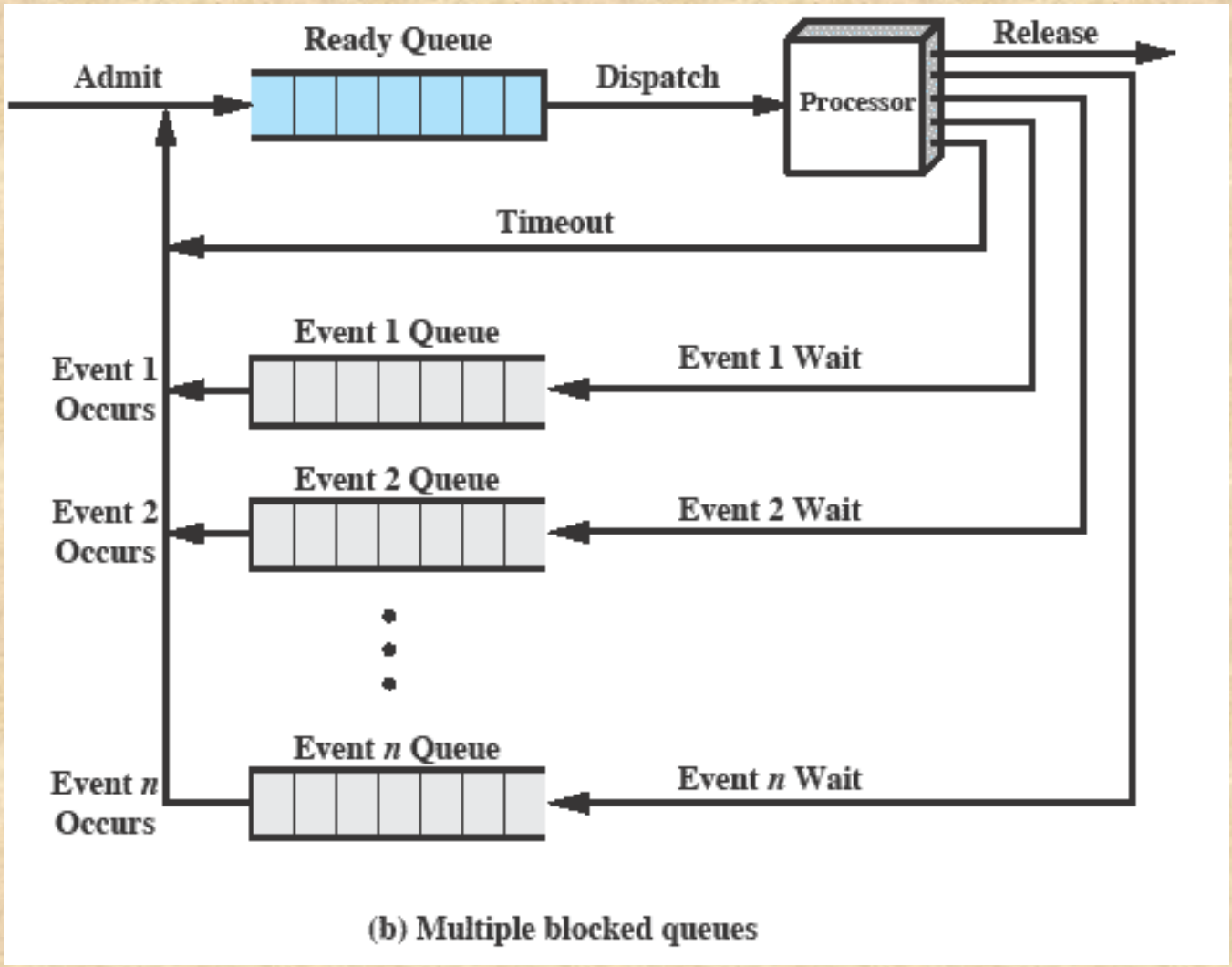
Figure 3.7 Process States for Trace of Figure 3.4

Upotreba dva reda čekanja



(a) Single blocked queue

Više redova čekanja



(b) Multiple blocked queues

Procesi i resursi

Šta OS treba da zna da bi upravljao procesima i resursima?

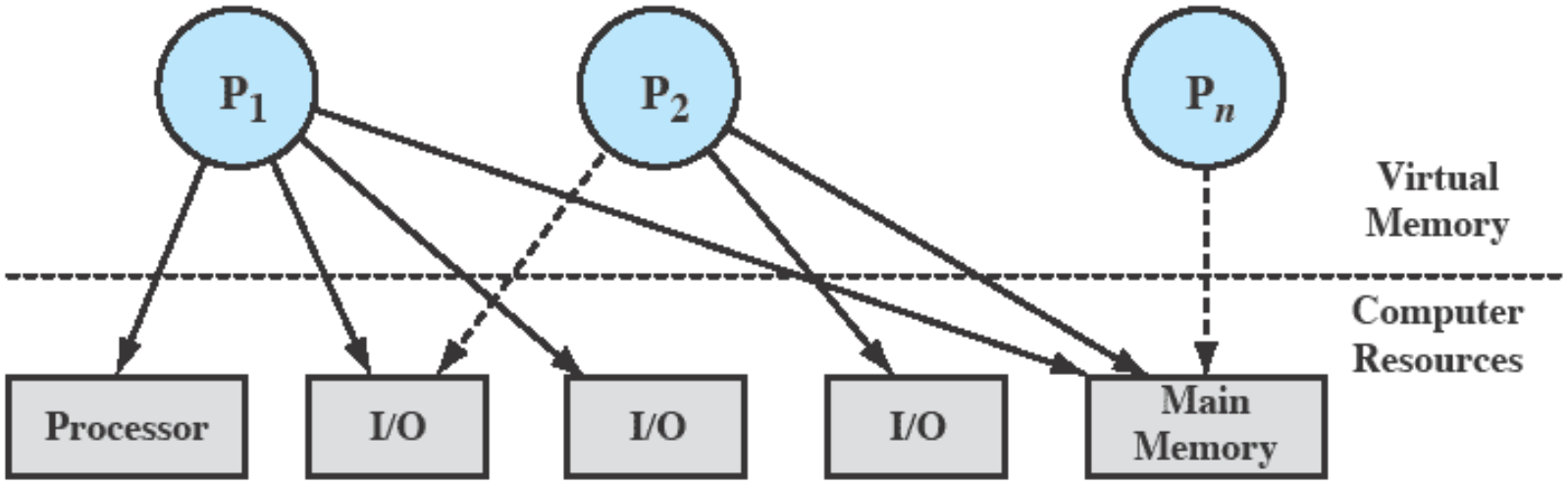


Figure 3.10 Processes and Resources (resource allocation at one snapshot in time)

Tabele OS-a



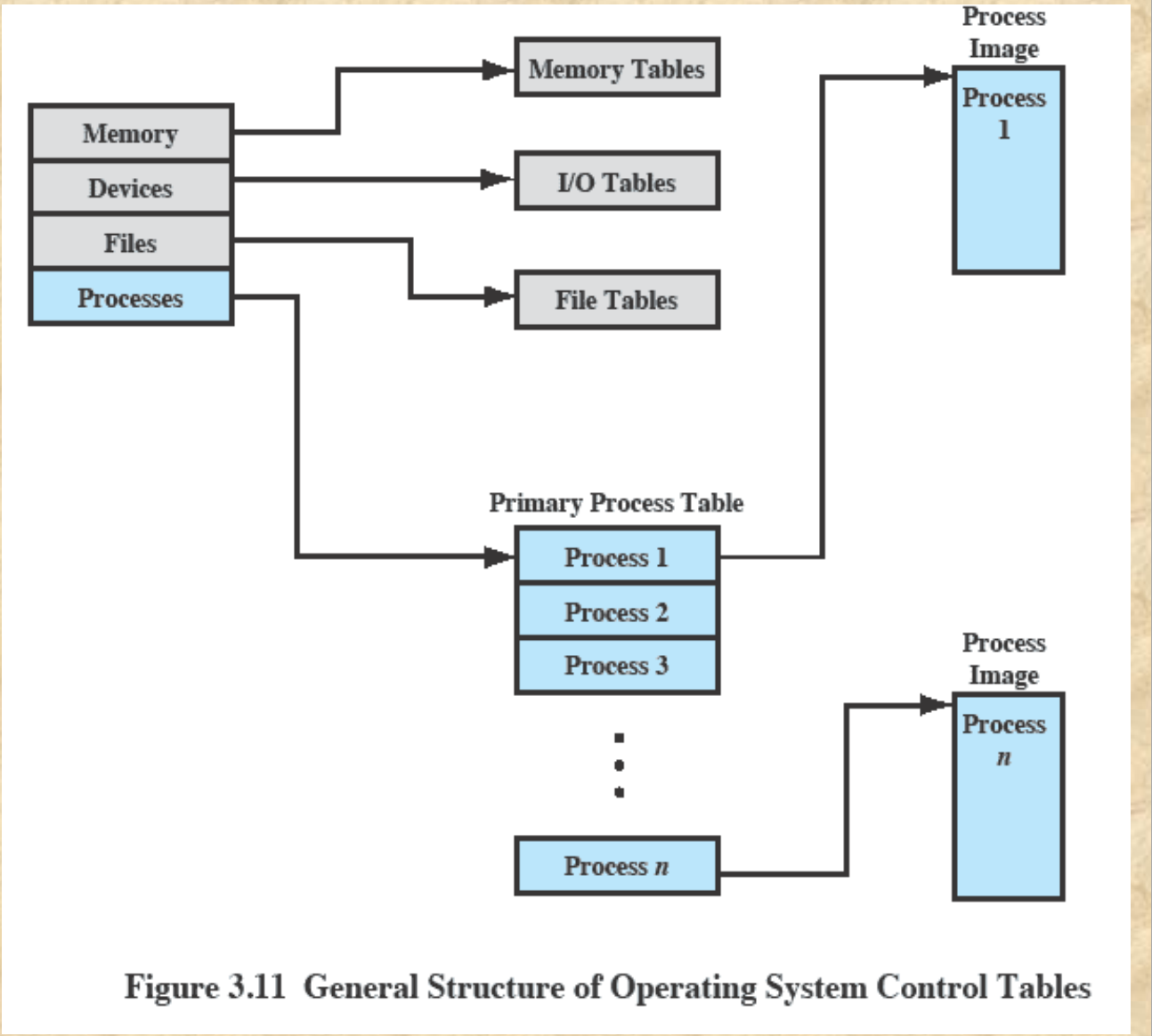


Figure 3.11 General Structure of Operating System Control Tables

Tabele memorije

- Služe za praćenje i glavne (realne) i sekundarne (virtuelne) memorije
- Procesi se čuvaju u sekundarnoj memoriji pomoću nekog oblika virtuelne memorije ili jednostavnog mehanizma razmene (swapping)

Obuhvataju:

- dodelu glavne memorije procesima
- dodelu sekundarne memorije procesima
- atribute zaštite blokova glavne ili virtuelne memorije
- informacije za upravljanje virtuelnom memorijom

U/I tabele

- OS ih koristi za upravljanje U/I uređajima i kanalima računarskog sistema
- U svakom trenutku U/I uređaj može biti slobodan ili dodeljen određenom procesu

Ako je U/I operacija u toku, OS treba da zna:

- status U/I operacije
- lokaciju u glavnoj memoriji koja se koristi kao izvor ili odredište U/I prenosa

Tabele datoteka

- Informacije može održavati i koristiti sistem za upravljanje datotekama ili sam OS

Ove tabele sadrže informacije o:

- postojanju datoteka
- lokaciji u sekundarnoj memoriji
- trenutnom statusu
- ostalim atributima

Tabele procesa

- Moraju se održavati radi upravljanja procesima
- Tabele procesa moraju sadržati reference na tabele memorije, U/I i datoteka
  - npr. pokazivače iz zapisa svakog procesa na njegove podatke u ostale tri grupe tabela.
- Same tabele moraju biti dostupne OS-u, pa i one podležu upravljanju memorijom
- Procesi su predstavljeni *slikom procesa*

Kontrolne strukture procesa - PCB

Komponente procesa

- Proces mora obuhvatati program ili skup programa koje treba izvršiti
- Proces obuhvata bar onoliko memorije koliko je potrebno za smeštanje programa i podataka tog procesa
- Izvršavanje programa obično uključuje stek koji služi za praćenje poziva procedura i prenosa parametara između procedura

Atributi procesa

- Svakom procesu pridružen je niz atributa koje OS koristi za kontrolu procesa – čuvaju se u procesnom kontrolnom bloku (PCB)
- Skup koji čine program, podaci, stek i atributi naziva se slika procesa
- Lokacija slike procesa zavisi od šeme upravljanja memorijom koja se koristi

Elementi PCB-a

Identifikacija procesa

Identifikatori

Numerički identifikatori koji se mogu čuvati u procesnom kontrolnom bloku obuhvataju

- Identifikator ovog procesa
- Identifikator procesa koji je kreirao ovaj proces (roditeljskog procesa)
- Identifikator korisnika

Informacije o stanju procesora

Registri vidljivi korisniku

Registar vidljiv korisniku je registar kome se može pristupiti iz mašinskog jezika koji procesor izvršava u korisničkom režimu. Obično ih ima od 8 do 32, mada neke RISC implementacije imaju više od 100.

Kontrolni i statusni registri

To su razni registri procesora koji služe za upravljanje radom procesora. Tu spadaju

- **Programski brojač:** Sadrži adresu sledeće instrukcije koju treba dohvatiti
- **Uslovni kodovi:** Rezultat poslednje aritmetičke ili logičke operacije (npr. znak, nula, prenos, jednakost, prekoračenje)
- **Statusne informacije:** Obuhvataju indikatore dozvoljenih/zabranjenih prekida, režim izvršavanja

Pokazivači steka

Svakom procesu pridružen je jedan ili više sistemskih stekova tipa LIFO (poslednji ušao, prvi izašao). Stek služi za čuvanje parametara i povratnih adresa pri pozivima procedura i sistemskim pozivima. Pokazivač steka pokazuje na vrh steka.

Elementi PCB-a (nastavak)



Informacije za upravljanje procesom

Informacije o raspoređivanju i stanju

Informacije koje su operativnom sistemu potrebne za raspoređivanje. Tipični podaci su:

- **Stanje procesa:** spremnost procesa da bude raspoređen na izvršavanje (npr. izvršava se, spreman, čeka, zaustavljen).
- **Prioritet:** jedno ili više polja koja opisuju prioritet procesa pri raspoređivanju; u nekim sistemima potrebno je više vrednosti (npr. podrazumevani, tekući, najviši dozvoljeni).
- **Informacije za raspoređivanje:** zavise od algoritma raspoređivanja, npr. koliko dugo proces čeka i koliko se dugo izvršavao poslednji put.
- **Događaj:** oznaka događaja na koji proces čeka da bi mogao da nastavi.

Struktuiranje podataka

Proces može biti povezan sa drugim procesima u red, prsten ili neku drugu strukturu. Na primer, svi procesi koji čekaju, a imaju isti nivo prioriteta, mogu biti povezani u red. Proces može biti u odnosu roditelj–dete (tvorac–stvoreni) sa drugim procesom. Procesni kontrolni blok može sadržati pokazivače na druge procese radi podrške ovim strukturama.

Međuprocesna komunikacija

Komunikaciji između dva nezavisna procesa mogu biti pridruženi razni indikatori, signali i poruke. Neke od tih informacija, ili sve, mogu se čuvati u procesnom kontrolnom bloku.

Privilegije procesa

Procesi dobijaju privilegije u pogledu memorije kojoj smeju da pristupaju i vrsta instrukcija koje smeju da izvršavaju. Privilegije se mogu odnositi i na korišćenje sistemskih uslužnih programa i servisa.

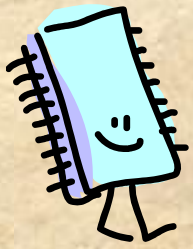
Upravljanje memorijom

Ovaj deo može sadržati pokazivače na tabele segmenata i/ili stranica koje opisuju virtuelnu memoriju dodeljenu procesu.

Vlasništvo nad resursima i njihovo korišćenje

Mogu biti navedeni resursi kojima proces upravlja, npr. otvorene datoteke. Može se čuvati i istorija korišćenja procesora i drugih resursa; te informacije mogu biti potrebne raspoređivaču.

Struktura slika procesa u virtuelnoj memoriji



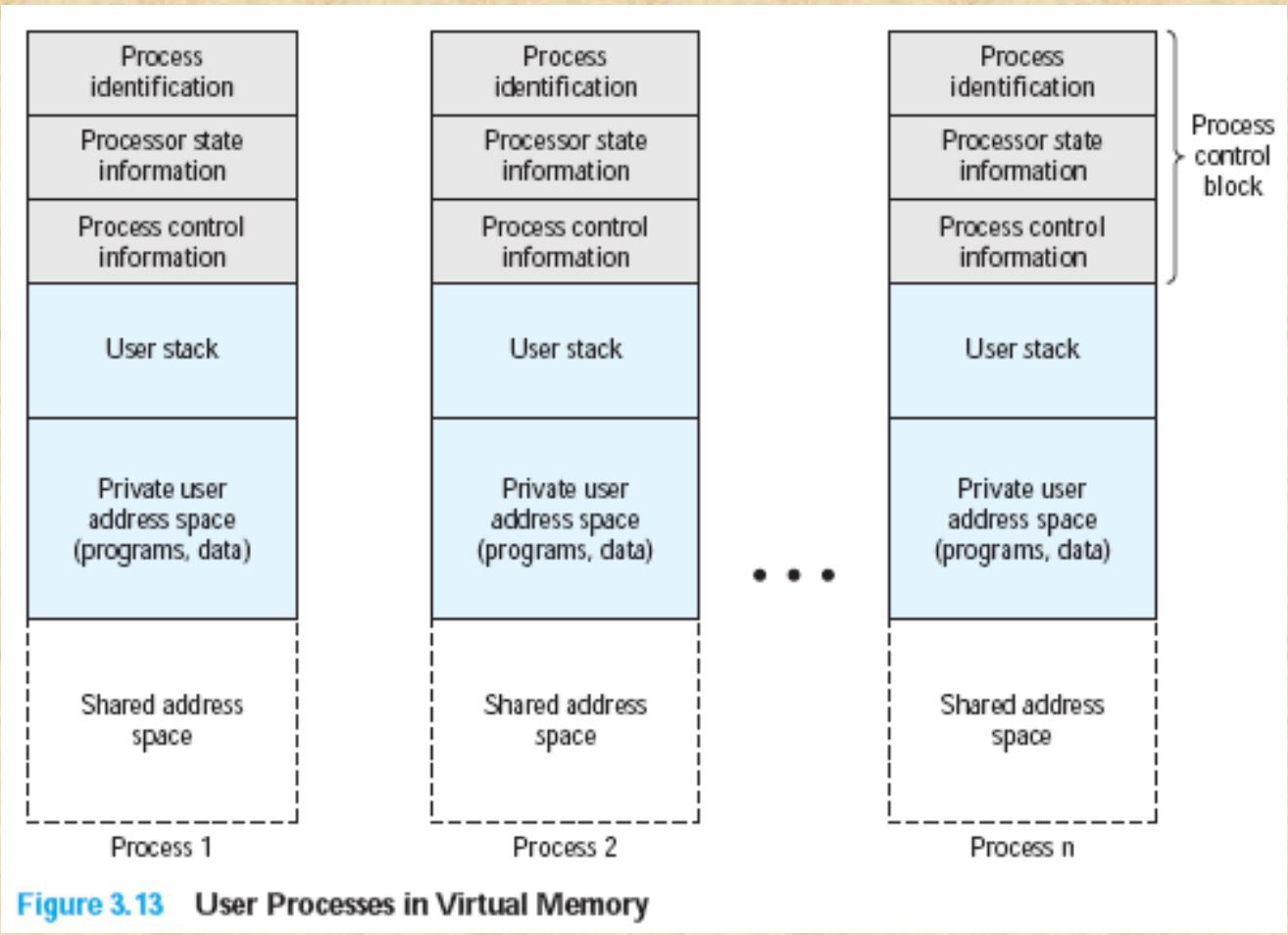


Figure 3.13 User Processes in Virtual Memory

Strukture lista procesa

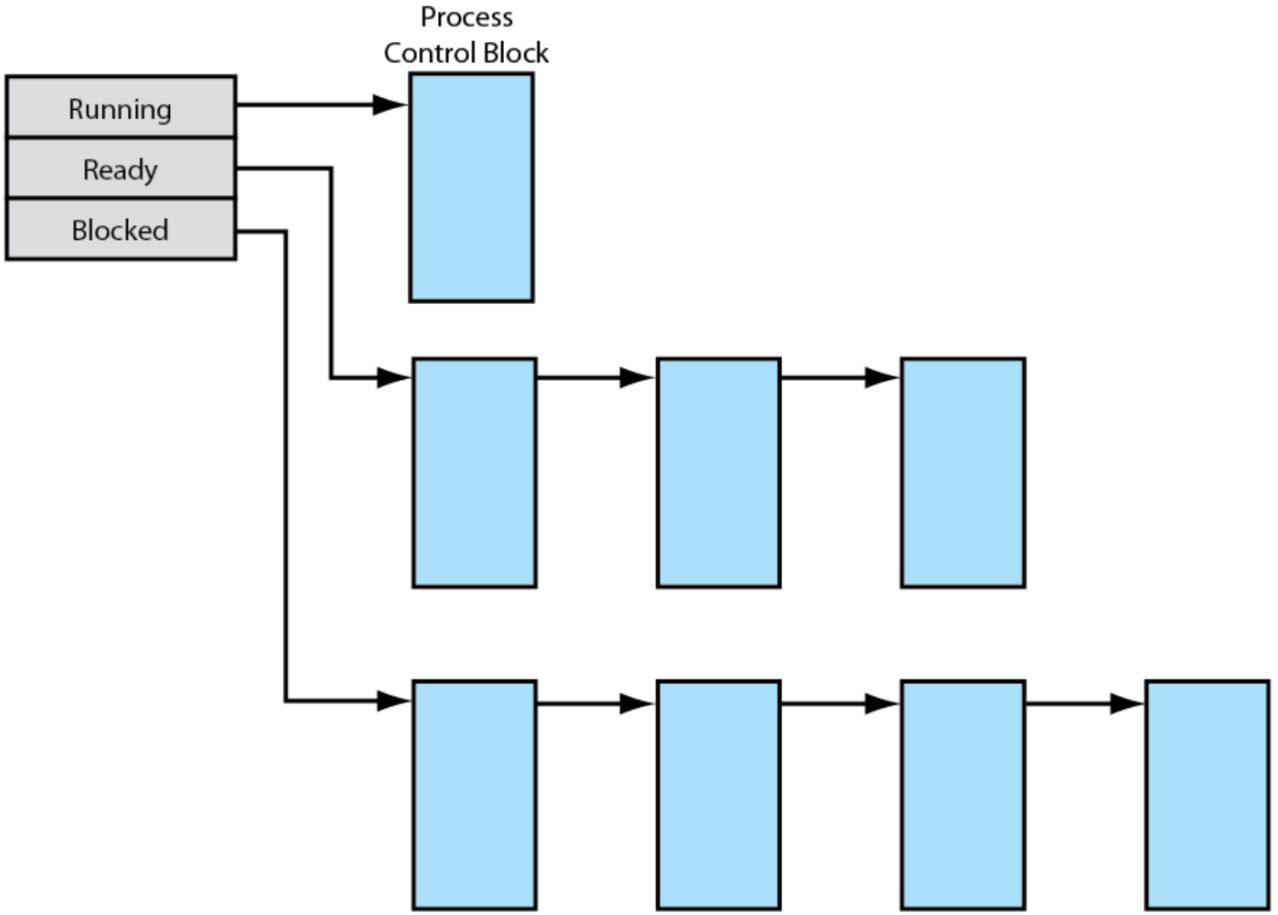


Figure 3.14 Process List Structures

Uloga procesnog kontrolnog bloka

- Najvažnija struktura podataka u OS-u
  - sadrži sve informacije o procesu koje su potrebne OS-u
  - blokove čita i/ili menja gotovo svaki modul OS-a
  - definiše stanje OS-a
- Problem nije pristup, već zaštita
  - greška u samo jednoj rutini može oštetiti procesne kontrolne blokove, što može onemogućiti sistem da upravlja pogođenim procesima
  - projektna izmena strukture ili semantike procesnog kontrolnog bloka može uticati na više modula OS-a

Režimi izvršavanja

Korisnički režim

- manje privilegovan
- korisnički programi se obično izvršavaju u ovom režimu



Sistemski režim

- privilegovaniji
- naziva se i kontrolni režim ili *režim jezgra*
- jezgro operativnog sistema



Tabela 3.7 Tipične funkcije koje obavlja jezgro OS-a

Upravljanje procesima

- Kreiranje i završetak procesa
- Raspoređivanje procesa i dodela procesora
- Promena procesa
- Sinhronizacija procesa i podrška za međuprocesnu komunikaciju
- Upravljanje procesnim kontrolnim blokovima

Upravljanje memorijom

- Dodela adresnog prostora procesima
- Razmena
- Upravljanje stranicama i segmentima

Upravljanje U/I

- Upravljanje baferima
- Dodela U/I kanala i uređaja procesima

Pomoćne funkcije

- Obrada prekida
- Obračun korišćenja resursa
- Nadgledanje

Kreiranje procesa

- Kada OS odluči da kreira novi proces, on:
  - dodeljuje novom procesu jedinstveni identifikator
  - alocira prostor za proces
  - inicijalizuje procesni kontrolni blok
  - postavlja odgovarajuće veze
  - kreira ili proširuje druge strukture podataka

Promena procesa (konteksta)

Promena procesa može nastupiti kad god OS preuzme kontrolu od procesa koji se trenutno izvršava. Događaji kojima OS dobija kontrolu su:

Mehanizam	Uzrok	Upotreba
Prekid	Spoljašnji u odnosu na izvršavanje tekuće instrukcije	Reakcija na asinhroni spoljašnji događaj
Zamka (trap)	Vezan za izvršavanje tekuće instrukcije	Obrada greške ili izuzetnog stanja
Sistemski poziv	Eksplicitan zahtev	Poziv funkcije operativnog sistema

Sistemski prekidi

Prekid

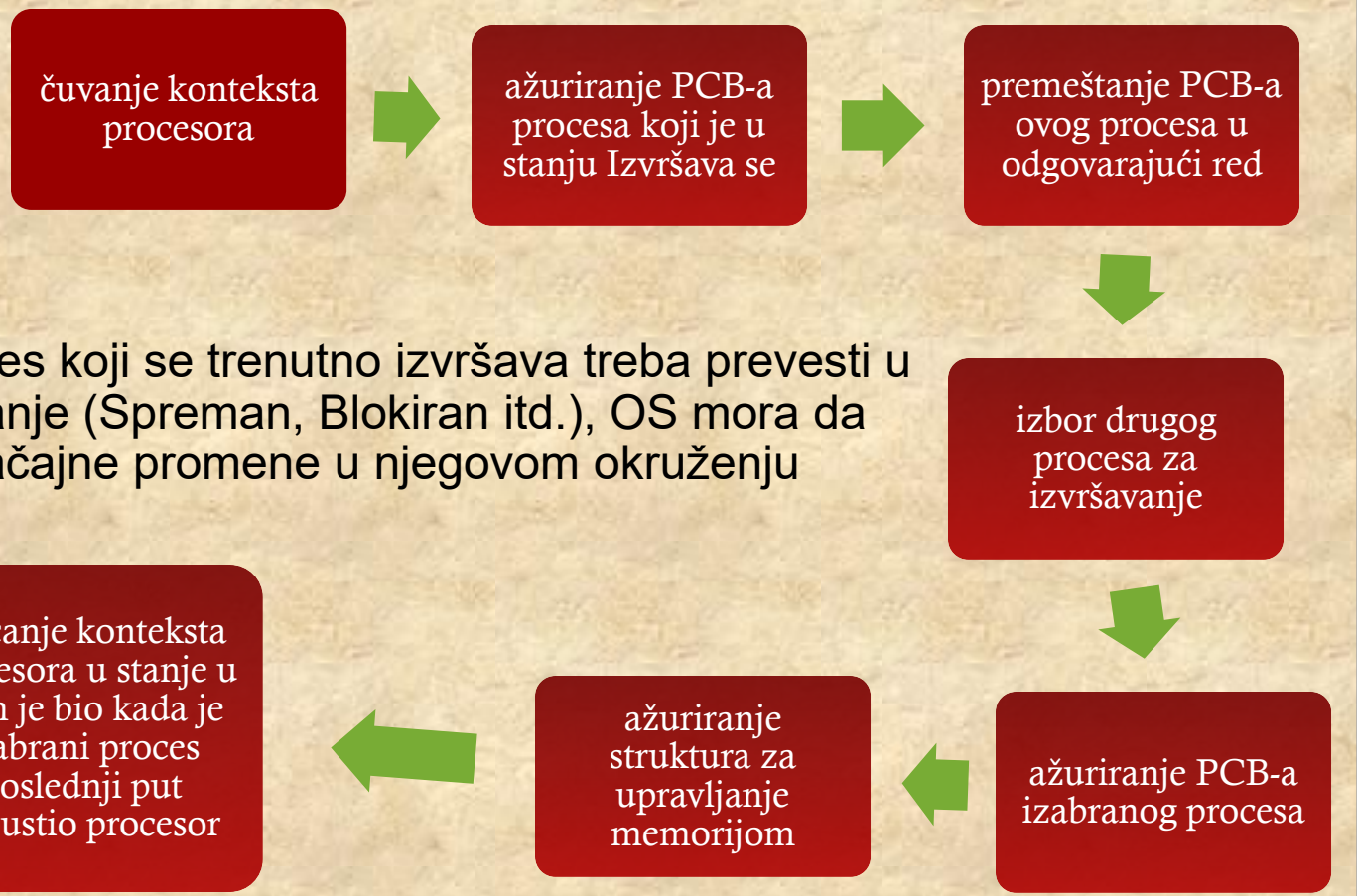
- Izaziva ga neki događaj koji je spoljašnji u odnosu na proces koji se trenutno izvršava i nezavisan od njega
  - prekid sata
  - U/I prekid
  - memorijska greška
- Vremenski kvantum
  - najduže vreme koliko proces može da se izvršava pre nego što bude prekinut

Zamka (trap)

- Greška ili izuzetno stanje nastalo u procesu koji se trenutno izvršava
- OS utvrđuje da li je stanje fatalno
  - proces prelazi u stanje Završen i dolazi do promene procesa
  - akcija zavisi od prirode greške

Promena stanja procesa

- Koraci potpune promene procesa su:



Ako proces koji se trenutno izvršava treba prevesti u drugo stanje (Spreman, Blokiran itd.), OS mora da izvrši značajne promene u njegovom okruženju

Pitanja bezbednosti

- OS svakom procesu pridružuje skup privilegija
- Proces koji se izvršava u ime korisnika obično ima privilegije koje OS priznaje tom korisniku
- Najviši nivo privilegija naziva se administratorski, supervizorski ili root pristup
- Ključni bezbednosni problem u projektovanju svakog OS-a je sprečiti, ili bar otkriti, pokušaje korisnika ili zlonamernog softvera da steknu neovlašćene privilegije u sistemu i root pristup



Pretnje pristupa sistemu

Uljezi

- Često se nazivaju hakerima ili krekerima

Klase:

- Maskirani uljez
- Nesavesni korisnik
- Prikriveni korisnik



- Cilj im je da dobiju pristup sistemu ili da prošire opseg privilegija koje imaju u sistemu
- Pokušavaju da dođu do informacija koje je trebalo da budu zaštićene

Zlonamerni softver

- Najsofisticiranije vrste pretnji računarskim sistemima

Kategorije:

- zavisni od programa domaćina (paraziti)
  - virusi, logičke bombe, zadnja vrata
- nezavisni
  - crvi, botovi

- Može biti relativno bezopasan ili vrlo štetan



Kontramere: detekcija upada

- „Bezbednosni servis koji prati i analizira sistemske događaje radi otkrivanja pokušaja neovlašćenog pristupa resursima sistema i upozoravanja na njih u realnom ili skoro realnom vremenu“ (RFC 2828)
- Može biti zasnovana na hostu ili na mreži
- Sistem za detekciju upada (IDS) sastoji se od tri logičke komponente:



- IDS-ovi se obično projektuju tako da otkrivaju ponašanje i ljudskih uljeza i zlonamernog softvera

Kontramere: autentifikacija

- „Proces provere identiteta koji neki entitet sistema tvrdi da ima ili koji mu se pripisuje.“ (RFC2828)
- Proces autentifikacije sastoji se od dva koraka:
  - Identifikacija
  - Verifikacija
- Četiri opšta načina autentifikacije korisnika:



Kontramere: kontrola pristupa

- Sprovodi bezbednosnu politiku koja određuje ko ili šta sme da pristupi svakom pojedinačnom resursu sistema i koja je vrsta pristupa dozvoljena u svakom slučaju
- Posreduje između korisnika i resursa sistema
- Administrator bezbednosti vodi bazu podataka o ovlašćenjima
- Funkcija revizije prati i evidentira pristupe korisnika resursima sistema



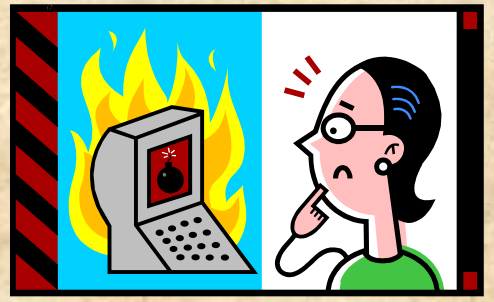
Kontramere: zaštitni zidovi

Namenski računar koji:

- predstavlja vezu sa računarima izvan mreže
- ima ugrađene posebne mere bezbednosti koje štite osetljive datoteke na računarima unutar mreže

Ciljevi zaštitnog zida:

- sav saobraćaj mora prolaziti kroz zaštitni zid
- propušta se samo ovlašćeni saobraćaj
- otporan na prodor



Stanja procesa u UNIX-u

User Running	Executing in user mode.
Kernel Running	Executing in kernel mode.
Ready to Run, in Memory	Ready to run as soon as the kernel schedules it.
Asleep in Memory	Unable to execute until an event occurs; process is in main memory (a blocked state).
Ready to Run, Swapped	Process is ready to run, but the swapper must swap the process into main memory before the kernel can schedule it to execute.
Sleeping, Swapped	The process is awaiting an event and has been swapped to secondary storage (a blocked state).
Preempted	Process is returning from kernel to user mode, but the kernel preempts it and does a process switch to schedule another process.
Created	Process is newly created and not yet ready to run.
Zombie	Process no longer exists, but it leaves a record for its parent process to collect.

Dijagram stanja procesa u UNIX-u

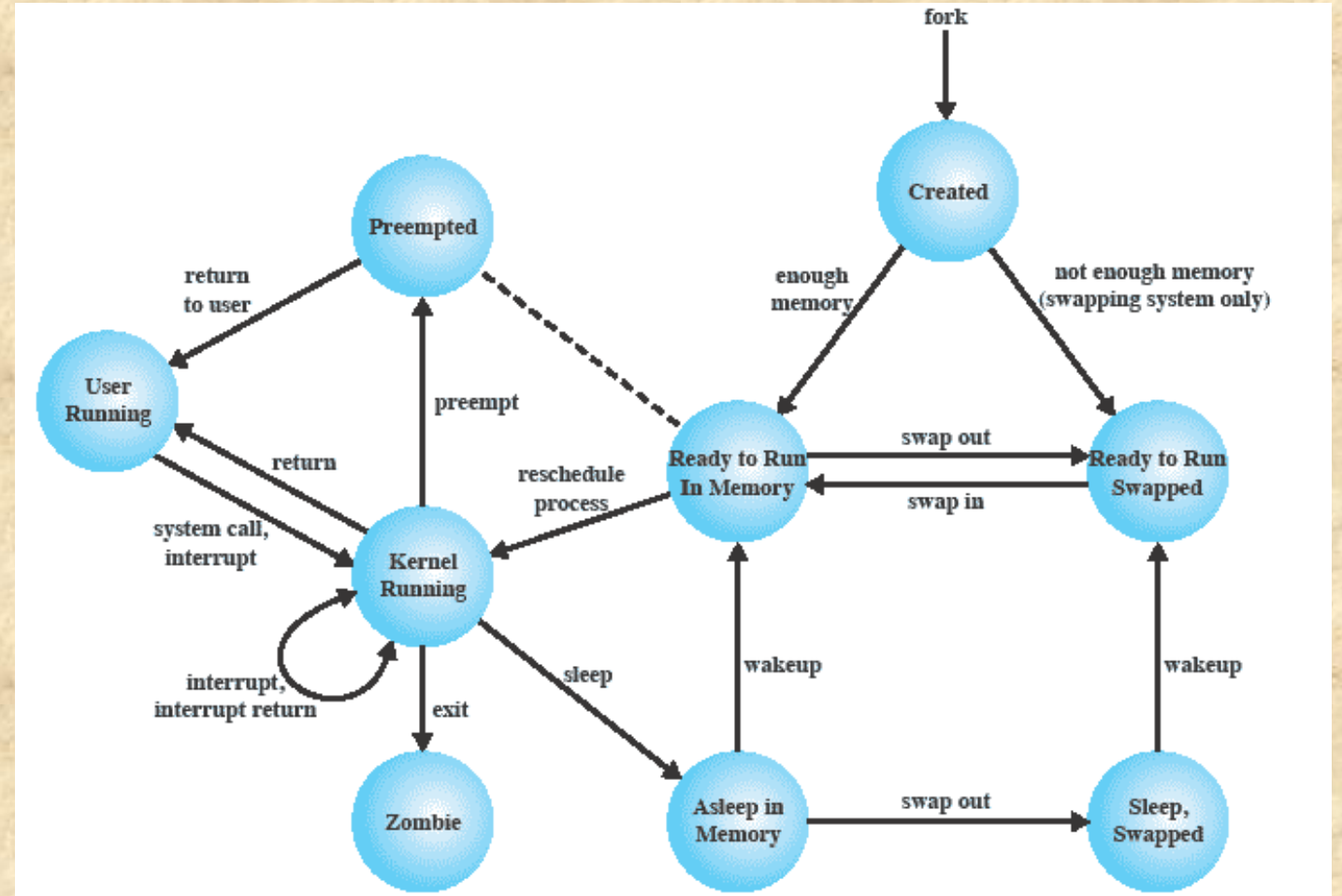


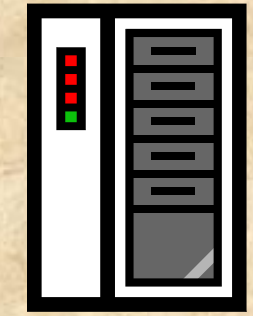
Figure 3.17 UNIX Process State Transition Diagram

Proces u UNIX-u



Kontekst na korisničkom nivou	
Tekst procesa	Izvršne mašinske instrukcije programa
Podaci procesa	Podaci dostupni programu ovog procesa
Korisnički stek	Sadrži argumente, lokalne promenljive i pokazivače za funkcije koje se izvršavaju u korisničkom režimu
Deljena memorija	Memorija deljena sa drugim procesima, koristi se za međuprocesnu komunikaciju
Kontekst registara	
Programski brojač	Adresa sledeće instrukcije koju treba izvršiti; može biti u memorijskom prostoru jezgra ili korisnika ovog procesa
Statusni registar procesora	Sadrži stanje hardvera u trenutku prekidanja; sadržaj i format zavise od hardvera
Pokazivač steka	Pokazuje na vrh steka jezgra ili korisničkog steka, u zavisnosti od režima rada u trenutku prekidanja
Registri opšte namene	Zavise od hardvera
Kontekst na sistemskom nivou	
Zapis tabele procesa	Definiše stanje procesa; ove informacije su uvek dostupne operativnom sistemu
U (user) oblast	Kontrolne informacije procesa kojima treba pristupati samo u kontekstu tog procesa
Tabela regiona procesa	Definiše preslikavanje virtuelnih adresa u fizičke; sadrži i polje dozvola koje pokazuje koja je vrsta pristupa dozvoljena procesu: samo čitanje, čitanje i upis ili čitanje i izvršavanje
Stek jezgra	Sadrži okvire steka procedura jezgra dok se proces izvršava u režimu jezgra

Tabela 3.11 Zapis tabele procesa u UNIX-u



Status procesa	Trenutno stanje procesa.
Pokazivači	Na U oblast i memoriju procesa (tekst, podaci, stek).
Veličina procesa	Omogućava operativnom sistemu da zna koliko prostora treba dodeliti procesu.
Identifikatori korisnika	**Stvarni ID korisnika** identifikuje korisnika odgovornog za proces koji se izvršava. **Efektivni ID korisnika** proces može da koristi da bi privremeno dobio privilegije određenog programa; dok se taj program izvršava kao deo procesa, proces radi sa efektivnim ID-om korisnika.
Identifikatori procesa	ID ovog procesa; ID roditeljskog procesa. Postavljaju se kada proces pređe u stanje Kreiran tokom sistemskog poziva fork.
Deskriptor događaja	Važeći kada je proces u stanju spavanja; kada se događaj desi, proces prelazi u stanje spreman za izvršavanje.
Prioritet	Koristi se za raspoređivanje procesa.
Signal	Nabraja signale koji su poslati procesu, a još nisu obrađeni.
Tajmeri	Obuhvataju vreme izvršavanja procesa, korišćenje resursa jezgra i korisnički tajmer za slanje signala alarma procesu.
P_link	Pokazivač na sledeći element u redu spremnih procesa (važeći ako je proces spreman za izvršavanje).
Status memorije	Pokazuje da li je slika procesa u glavnoj memoriji ili je izbačena na disk. Ako je u memoriji, ovo polje pokazuje i da li sme biti izbačena ili je privremeno zaključana u glavnoj memoriji.

Kreiranje procesa

- Proces se kreira sistemskim pozivom jezgra fork( )
- Tada OS, u režimu jezgra:

1. Rezerviše mesto u tabeli procesa za novi proces
2. Dodeljuje jedinstveni ID procesu detetu
3. Pravi kopiju slike roditeljskog procesa, osim deljene memorije
4. Uvećava brojače za sve datoteke čiji je vlasnik roditelj, jer ih sada poseduje i još jedan proces
5. Prebacuje proces dete u stanje Spreman za izvršavanje
6. Vraća ID deteta roditeljskom procesu, a vrednost 0 procesu detetu

Posle kreiranja

- Posle kreiranja procesa, jezgro u okviru rutine dispečera može da uradi jedno od sledećeg:
  - ostati u roditeljskom procesu
  - preneti kontrolu na proces dete
  - preneti kontrolu na neki drugi proces



Rezime

- Najosnovniji koncept savremenog OS-a je proces
- Glavna funkcija OS-a je da kreira procese, upravlja njima i okončava ih
- Procesni kontrolni blok sadrži sve informacije koje su OS-u potrebne za upravljanje procesom, uključujući njegovo trenutno stanje, resurse koji su mu dodeljeni, prioritet i druge relevantne podatke
- Najvažnija stanja su Spreman, Izvršava se i Blokiran
- Proces koji se izvršava je onaj koji procesor trenutno izvršava
- Blokirani proces čeka na završetak nekog događaja
- Proces koji se izvršava prekida se prekidom ili izvršavanjem sistemskog poziva OS-u