

Komunikacija između procesa

- Kod multiprogramskog rada, **složeni poslovi se po pravilu dele na manje module** koji su jednostavniji, pregledniji, lakši za nalaženje grešaka, a uz to su i **tako organizovani da imaju mogućnost paralelnog izvršavanja** (vezani su za različite resurse i periferije). Svaki od njih pojedinačno **predstavlja jedan proces**.
- U cilju razmene podataka između procesa, uvode se **odgovarajući i mehanizmi** koji obezbeđuju njihovu međusobnu komunikaciju.
- U osnovi postoje dva načina da se to izvede:
 - putem **deljive memorije**, i
 - putem **razmene poruka**.

Komunikacija između procesa

- Komunikacija putem deljive memorije zahteva od procesa da poseduju **zajednički deo memorije.**
- **Izbor tipa i realizacija komunikacije** je prepuštena autoru aplikacije.
- Operativni sistem je odgovoran **samo za obezbeđenje zajedničke memorije.**

Komunikacija između procesa

- **Kod metoda razmene poruka** komunikacija se odvija posredstvom **operativnog sistema**, odnosno **sistemskih poziva** za razmenu poruka.
- To je u mnogo slučajeva od velikog značaja jer se **korisnik oslobađa razrešavanja brojnih pitanja**, koja se mogu javiti pri izboru načina komunikacije između procesa:
 - kako se uspostavlja veza između procesa ?
 - može li se veza uspostaviti između više od dva procesa?
 - koliko veza može postojati između svakog para procesa?
 - postoji li bafer i koliki je kapacitet veze ?
 - da li su poruke fiksne ili promenjive dužine ?
 - da li je veza između procesa jednosmerna ili dvosmerna?

Komunikacija između procesa

- **Zavisno od izbora projektanta** operativnog sistema (odnosno programera ukoliko sam definiše komunikaciju između procesa), **razlikujemo više tipova komunikacije:**
 - **direktna ili indirektna komunikacija** (poruke se razmenjuju direktno između procesa ili se šalju u posebna područja zvana **poštanski sandučići** - procesi komuniciraju ako imaju zajednički poštanski sandučić), tj. radi se ili o slanje direktno procesu ili u poštanski sandučić;
 - **simetrična ili asimetrična komunikacija** (da li je veza dvosmerna ili jednosmerna);
 - **automatsko ili eksplicitno baferovanje** (bez bafera, sa konačnim baferom i sa beskonačnim baferom);
 - **slanje kopije ili pokazivača** (pointera);
 - **fiksna ili promenljiva dužina poruke**, itd.

Komunikacija između procesa

- Svaka od ovih metoda ima svoje **prednosti i mane** od kojih nisu sve na prvi pogled vidljive.
- Stoga je **za korisnika mnogo bolje** ukoliko brigu vezanu za izbor i eventualne probleme koji se mogu pojaviti, **preuzme operativni sistem** odnosno njegov projektant.
- Većina operativnih sistema ima već definisane na interne međusobne komunikacije i njima pripadajuće strukture.
- Tako već **mnogi operativni sistemi pružaju mogućnost definisanja maksimalnog vremena** u kojem se podaci moraju razmeniti.
- **Nakon isteka ovog vremena** javlja se poruka greške procesu koji otključuje podatak.

Komunikacija između procesa

- Operativni sistemi za rad u realnom vremenu naj češće koriste sledeće mehanizme za komunikaciju bazirane na razmeni poruka:
 - Mehanizam poštanskog sandučeta ("mailbox"),
 - Mehanizam direktne razmene poruka ("message") i
 - Mehanizam kružnog bafera ("queue").

Mehanizam poštanskog sandu eta ("mailbox")

- **Mehanizam poštanskog sandu eta** koristi posebnu strukturu za razmenu poruka koja se naziva poštansko sandu e.
- **Veli ina poštanskog sandu eta zavisi od procesora i njemu pripadaju eg operativnog sistema**, ali se uvek bira tako da se u njoj može smestiti **pokaziva (pointer)** kojim se može obuhvatiti kompletan adresni prostor (ili onaj deo memorije koji je predvi en za razmenu poruka).

Mehanizam poštanskog sandu eta ("mailbox")

- Pred operativnim sistemima, koji poseduju ovu strukturu kao zadatak se name e **razrešenje dve slede e situacije:**
 - šta uraditi ako u "mailbox" **istovremeno pristignu dve poruke** (ili za vreme obrade jedne od njih pristigne druga), a da se pri tome ne izgubi ni jedna od njih?
 - šta uraditi ako **neki od procesa traži poruku od datog procesa za produžetak svog rada?** Ili preciznije šta ako to isto zahteva više procesa istovremeno?

Mehanizam poštanskog sandu eta ("mailbox")

- Ovaj problem neki operativni sistemi razrešavaju time što **pri kreiranju "mailbox"-a generišu i dve liste** - jednu sa procesima koji upisuju poruke, a drugu sa procesima koji ekaju prispe e poruke (ukoliko je "mailbox" prazan).
- Pri tome se **novi procesi** u liste ubacuju na osnovu jednog od slede a dva kriterijuma - **na osnovu prioriteta** pošiljaoca **ili po FIFO kriterijumu**.

Mehanizam poštanskog sandučeta ("mailbox")

- **Drugi operativni sistemi, umesto ulazne liste** koriste jednu drugu listu organizaciju - u samoj poruci nalazi se polje koje služi za međusobno povezivanje poruka.
- Umesto se **i lista procesa koji čekaju na poruku** izbacuje, a umesto nje se u PCB-u ubacuje polje kojim se isti povezuju.
- Najjednostavniji metod je da se poruke uopšte i ne povezuju, već da se njihova obrada **prepusti korisniku**.
- U slučaju da **neka poruka stigne a da prethodna nije** poslata, pošiljaocu poruke se **posle stigne poruka o grešci** a programeru ostavi eventualna obrada greške.

Mehanizam poštanskog sandueta ("mailbox")

Struktura kojoj se pristupa preko sistemskih poziva za upis i čitanje.

- Sistemski poziv za upis u "mailbox", **upisuje vrednost pokazivača na poruku** u "mailbox".
- **Ukoliko već postoji prisutna poruka koja još nije obrađena tada se vrši povezivanje sa prethodnom porukom na već pomenuti način.**
- U slučaju da postoji **proces koji je suspendovan i čeka na prijem poruke**, vrši se njegovo prevođenje u stanje spreman i predaje mu se pokazivač na poruku. **Proces se skida iz reda** odnosno liste procesa koji čekaju na poruku. Nakon ovoga se **poziva dispatcher** koji pokreće proces sa najvećim prioritetom.

Mehanizam poštanskog sandučeta ("mailbox")

- Sistemska poziva za čitanje "mailbox"-a **ukoliko je isti prazan vrši suspenziju procesa sve do prispeća poruke**. Broj procesa se stavlja u listu za čitanje, a nakon toga poziva disperziju.
- **Ukoliko "mailbox" nije prazan** (postoji pokazivač na poruku u njemu), **procesu se predaje njegov sadržaj**, a sadržaj "mailbox-a" se **isti i u njega upisuje pointer na sledeću poruku**.

Mehanizam direktne razmene poruka ("message")

- Poruka je **struktura fiksne ili promenljive dužine** koja se koristi **za razmenu podataka** između dva (ili više) procesa.
- Poruka pored **delo u kome su smešteni podaci** najčešće sadrži i **takozvano zaglavlje poruke**. U zaglavlju se smeštaju **polja koja operativni sistem koristi** za svoje potrebe.
- To su obično dužina poruke, adrese odredišnog procesa i procesa pošiljaoca, polje za povezivanje poruka u lanac (linkovanje) i neka komandna polja (slika 3.9.).

Mehanizam direktne razmene poruka ("message")



Slika 3.9 Struktura poruke ("message").

Mehanizam direktne razmene poruka ("message")

- Svrha polja za povezivanje je u tome što se izuzetno **retko** razmena poruka vrši **kopiranjem** njenog sadržaja, **ve** se to ini **dostavljanjem njene adrese** (pokaziva a).
- U slu aju da isti proces **dobije više poruka** **one se povezuju u liste** time što prethodna poruka u polju za povezivanje nosi adresu slede e.

Mehanizam direktne razmene poruka ("message")

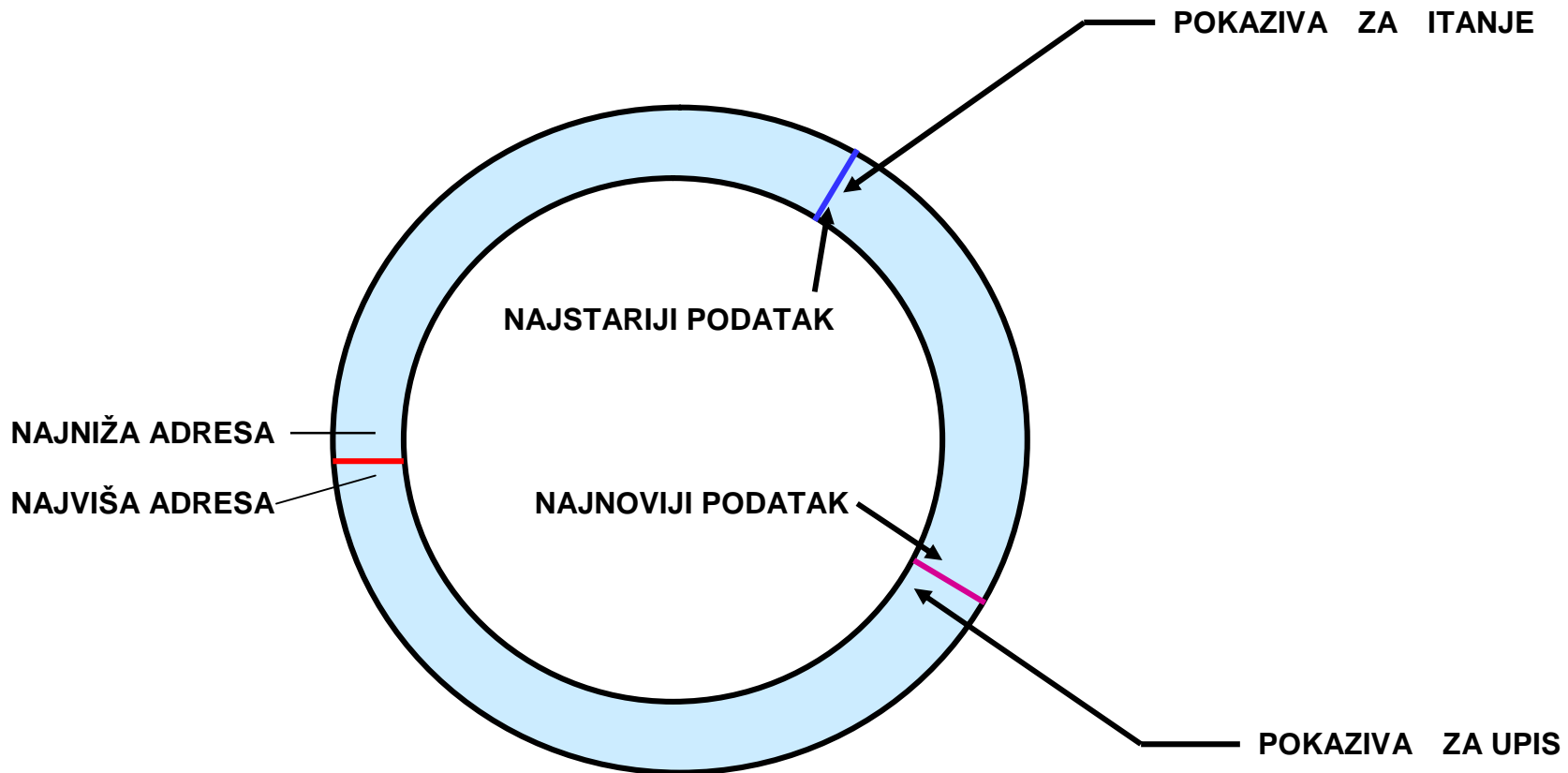
- Adresa prve poruke u listi se naj eš e nalazi u **Kontrolnom bloku procesa (PCB)**, a poslednja poruka u nizu u svom polju za povezivanje ima **oznaku kraja liste** (nulu ili vrednost koja nikako ne može biti adresa neke poruke).
- Kako se prenošenje podataka putem poruka vrši bez kopiranja sadržaja, to je **ovaj metod komunikacije izuzetno brz**.
- Iz datog razloga **koristi se u svim slu ajevima** u kojima je neophodno obezbediti **brz protok informacija izme u dva procesa**.

Mehanizam kružnog bafera ("queue")

- Kružni bafer (slika 3.10.) je struktura fiksne (i kona ne) dužine namenjena razmeni podataka izme u procesa.
- Sastoji se od polja za smeštaj podataka (kona ne dužine) i dva pokaziva a -jednog za upis i drugog za itanje.
- Pokaziva za upis pokazuje na prvu slobodnu lokaciju za upis podatka, a pokaziva za itanje pokazuje na prvi nepro itani podatak.
- Dolaskom do kraja prostora za smeštaj podataka i jedan i drugi pokaziva se ponovo vra aju na po etak.

Mehanizam kružnog bafera ("queue")

Kružni bafer



Slika 3.10. Kružni bafer

Mehanizam kružnog bafera ("queue")

- Bafer je prazan kada se pokazivač očitanih podataka izjednači sa pokazivačem upisanih.
- Bafer je pun i ne može da primi podatke bez njihovog prepisivanja, kada pokazivač upisa dostigne vrednost pokazivača očitanih podataka (napravi pun krug).
- U ovom slučaju **običajno se javlja poruka greške** a upis se zabranjuje ili dozvoljava zavisno od načina realizacije rutina koje ovaj posao obavljaju.
- Podaci u kružni bafer **mogu biti upisivani od strane više procesa**, a isto tako i očitavani.
- Mnogo je slučajeva da upis u kružni bafer vrši jedan proces, a očitavanje takode samo jedan proces.

Mehanizam kružnog bafera ("queue")

- **Ukoliko dva ili više procesa uzimaju podatke iz kružnog bafera, tada se pri odlučivanju koji će od njih prvo dobiti podatak koriste dva kriterijuma.**
- **Prvi je FIFO, a drugi je na osnovu prioriteta - podatak dobija proces sa većim prioritetom.**
- **Koji će od ova dva kriterijuma biti primenjen zavisi od operativnog sistema.**
- **Neki operativni sistemi dozvoljavaju mogućnost izbora - kriterijum se navodi pri kreiranju ove strukture.**

Mehanizam kružnog bafera ("queue")

- S obzirom da se **transfer podataka vrši kopiranjem** i da se za svaki preneseni podatak pozivaju **najmanje dva systemska poziva (za upis i čitanje podatka)**, **prenos podataka je sporiji** nego pri prenosu poruka.
- Iz tog razloga se ova struktura izbegava u slučajevima kada je potrebna velika brzina prenosa podataka.
- Osobina ove strukture jeste da **čuva hronološki redosled događaja**.
- Zbog toga se **često koristi za prihvatanje (ili izdavanje) podataka tipa niza karaktera** koji stižu (ili se izdaju) asinhrono, kao i za druge tipove baferisanja.

Mehanizam kružnog bafera ("queue")

- Ovoj strukturi se pristupa **preko sistemskih poziva za čitanje i upis.**
- Pozivi za upis u kružni bafer vrše **upis podatka na kraj liste i pomeraju pokazivača** za upis na prvu slobodnu lokaciju za jedno mesto (FIFO redosled upisa).
- Ukoliko nema slobodnog mesta, tada se **prepisuje podatak i javlja poruka o grešci** ili što je mnogo češće **suspenduje se proces pošiljalac** sve do oslobađanja prostora za smeštaj podatka.

Mehanizam kružnog bafera ("queue")

- Pozivi za čitanje iz kružnog bafera vrše čitanje podatka sa vrha liste i pomeraju pokazivača preko čitanih podataka za jedno mesto.
- Ukoliko nema podatka u kružnom baferu, tada se proces koji je uputio poziv suspenduje i kontrola predaje dispečeru.
- U slučaju da postoje više procesa koji čekaju na podatak iz istog kružnog bafera, tada se u trenutku upisa ovaj proces uje ili procesu koji najduže čeka na podatak ili procesu sa najvećim prioritetom.

Sinhronizacija procesa

- **Procesi u multiprogramskom okruženju** i pored toga što predstavljaju zaokružene celine, **ne postoje sami za sebe**, već je njihova uslovljenost veoma često višestruka i raznovrsna.
- Da bi videli **kakvi se sve problemi mogu javiti u pogledu odnosa između procesa** iznećemo nekoliko karakterističnih primera:
 - **dva ili više procesa mogu da dele određeni resurs**, pri čemu istovremeno samo jedan od njih može isti koristiti (na primer zajednički komunikacioni kanal);

Sinhronizacija procesa

- proces se može izvršavati tek po izvršenju drugog (ili više njih), ta no odre enog procesa (npr. obrada rezultata merenja se ne može izvršiti pre samog merenja);
- proces se izvršava tek po izvršenju odre enog procesa ije se izvršavanje suspenduje sve dok se neki dati proces ne izvrši (mehanizam klackalice);
- proces postaje aktivan dešavanjem bilo kog od više doga aja;
- više procesa postaje spremno kompletiranjem odre enog procesa;
- proces se suspenduje do isteka nekog vremenskog intervala.

Sinhronizacija procesa

- Naj češće korišćene strukture za sinhronizaciju su:
 - mehanizam **semafora** ("semaphor"), i
 - mehanizam **dogaja** ("flaggroup").
- Pored ovih struktura za sinhronizaciju procesa **se mogu koristiti i strukture za komunikaciju između procesa.**
- **Slanje poruke bez sadržaja (ili sa proizvoljnim sadržajem) u poštansko sanduće ili putem mehanizma direktne razmene poruka može se iskoristiti za signalizaciju određenog dogaja i sinhronizaciju dva ili više procesa.**

Sinhronizacija procesa

- Operativni sistemi **esto ne poseduju sve navedene strukture** (ili ne u njihovoj izvornoj definiciji).
- Razlog za takvo stanje je u injenici da se **rad pojedinih struktura može simulirati** korišćenjem postojećih struktura za sinhronizaciju i uz malo programerskog truda.
- Neki operativni sistemi omogućavaju i **definisanje maksimalnog vremena u kojem se mora desiti sinhronizacija** sa drugim procesom u cilju sprečavanja **trajnog blokiranja** procesa.

Mehanizam semafora ("semaphore")

- **Semafor** je celobrojna promenljiva, čiji se sadržaj koristi za sinhronizaciju između procesa na sledeći način:
 - **Procesu se dozvoljava nastavak izvršavanja samo ako je sadržaj semafora pozitivan** ili ima određenu pozitivnu vrednost.
 - **U suprotnom proces se suspenduje** sve dok semafor ne dostigne potrebnu vrednost.
- S obzirom da **više procesa mogu pristupati istom semaforu** ovoj strukturi **se pridružuje i lista procesa koji čekaju na njega** (ili se suspendovani procesi ulančavaju).
- **Procesi koji čekaju, u listu se ubacuju** ili po FIFO ili po kriterijumu prioriteta.

Mehanizam semafora ("semaphore")

- Ova struktura omogućava da u slučaju da se događaj javi više puta, svaka njegova pojava ostane zabeležena.
- Time se omogućava procesu koji čeka na ovu strukturu, isti toliki ili odgovarajući broj izvršavanja, u slučaju da dati proces zbog niskog prioriteta ili iz nekog drugog razloga nije bio u stanju da dobije procesor u trenutku signalizacije događaja.
- Strukturi se pristupa preko sistemskih poziva za signalizaciju(signal) i čekanje (wait) na semafor.

Mehanizam semafora ("semaphore")

- Signalizacija semaforu predstavlja inkrementiranje njegovog sadržaja odnosno povećanje njegove vrednosti za definisani ceo broj.
- Pri tome **systemski poziv obuhvata** sledeće akcije:
 - ukoliko ni jedan proces ne čeka na semafor **isti se inkrementira**;
 - ukoliko postoji proces (ili više njih), koji čeka na dati semafor isti se prevodi u stanje spreman, a zatim se vrši **ažuriranje semaforu pridružene liste**. U ovom slučaju se **ne vrši inkrementiranje semafora**.

Mehanizam semafora ("semaphore")

- Poziv za čekanje na semafor obuhvata sledeće akcije:
 - **ukoliko je vrednost semafora veća od nule, vrši se njeno dekrementiranje** a proces koji je uputio poziv **nastavlja sa izvršavanjem**;
 - **ukoliko je sadržaj semafora manji ili jednak nuli**, proces se **suspenduje i ubacuje u listu procesa koji čekaju na dati semafor**. Nakon toga **poziva se dispatcher** koji pokreće proces **najvećeg prioriteta**.

Mehanizam događaja ("flaggroup")

- Mehanizam događaja se koristi u višestrukoj **sinhronizaciji** tj. pri:
 - sinhronizaciji **jednog događaja sa više drugih**, i
 - sinhronizaciji **više događaja sa jednim**.
- Najčešće se ostvaruje preko strukture od celog broja bajtova u kojoj **svaki bit predstavlja poseban fleg odnoso događaj**.
- Fleg može imati jedno od dva stanja: **setovan ("1")**
 - događaj se desio i **obrisan ("0")** - događaj se još **nije desio**.

Mehanizam događaja ("flaggroup")

- Pri tome mogu biti obezbeđena **ena sledeća dva tipa sinhronizacije:**
 - **disjunktivna** ("OR" - "ILI") pri kojoj proces postaje spreman **pri pojavi prvog od niza** specificiranih događaja;
 - **konjunktivna** ("AND" - "I") pri kojoj proces postaje spreman tek **pri pojavi svih** specificiranih događaja.
- S obzirom da više procesa istovremeno mogu **ekati pojavu određenog događaja**, time se omogućava da **jedan proces može svima njima da pošalje obaveštenje** da se neki događaj desio.

Mehanizam događaja ("flaggroup")

- Korisnik ovoj strukturi pristupa **preko sistemskih poziva: za setovanje flega, za čekanje na fleg i brisanje flega.**
- Poziv za setovanje flega, postavlja jedan ili više flegova definisanih maskom na jedinicu.
- **Procesi koji su čekali** na date flegove **postaju spremni za izvršavanje**, a upravljanje se predaje **dispečeru.**
- **Ukoliko su neki od flegova već setovani, ostali se setuju a procesu se vraća poruka greške.**

Mehanizam događaja ("flaggroup")

- Pozivom za čekanje na flag, specificiraju se događaji na koje proces čeka i da li to radi disjunktivno ili konjuktivno.
- Ako su se traženi događaji već desili, proces nastavlja izvršavanje.
- U suprotnom proces se suspenduje i poziva disper.

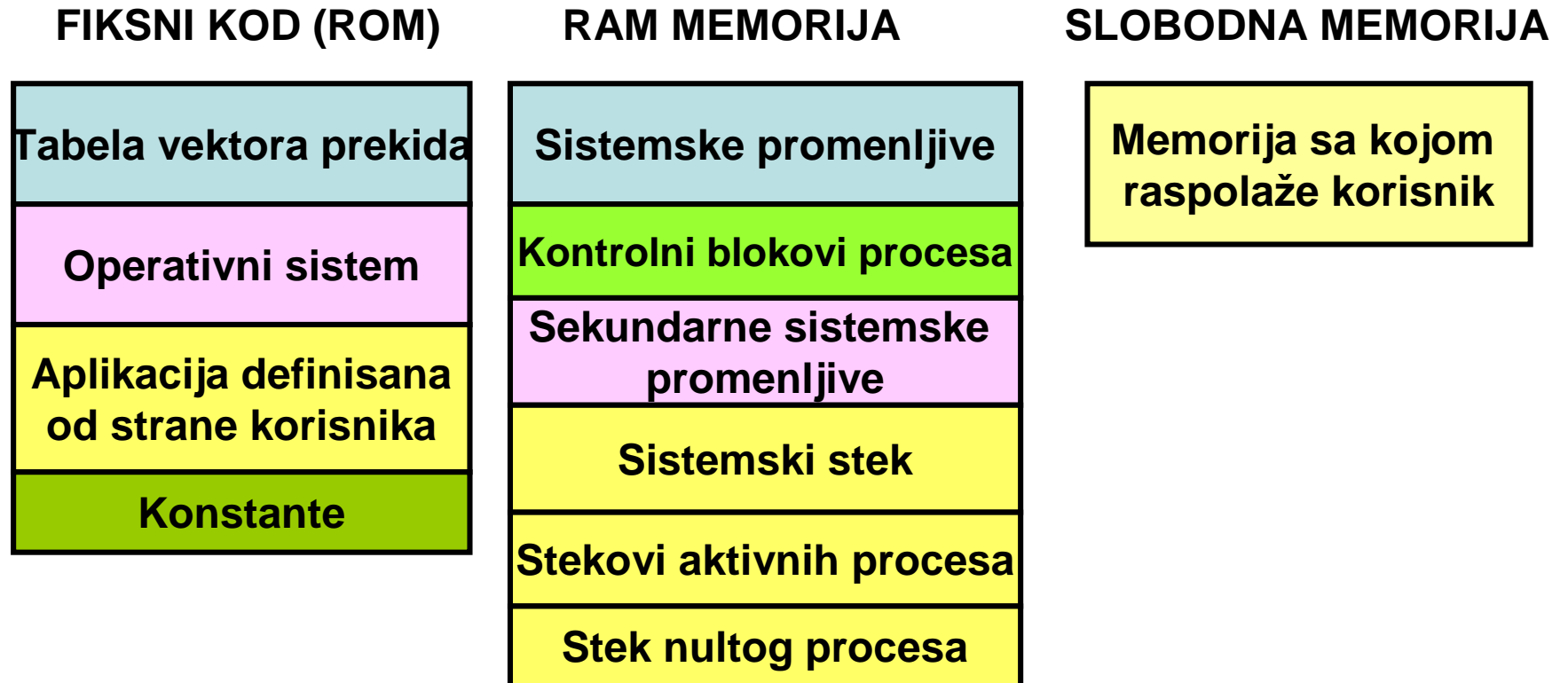
Mehanizam događaja ("flaggroup")

- Pozivom za brisanje, flegovi specificirani maskom se postavljaju na nulu.
- Ukoliko neki od specificiranih flegova ve ima vrednost nula, procesu koji je inicirao brisanje **vraća se poruka greške.**

Organizacija memorije

- U okviru adresnog prostora svakog procesora **smeštaju se program i podaci** sa kojima procesor trenutno radi.
- U **multiprogramskom okruženju** u memoriji se nalazi **više programa i njima pripadaju ih podataka** kao i program koji omogućava multiprogramski rad - **operativni sistem** sa svojim internim (sistemskim) promenljivim.
- U opštem slučaju organizacija i sadržaj memorije izgleda kao na slici 3.11.

Organizacija memorije



Slika 3.11 Sadržaj memorije multiprogramskog operativnog sistema.

Organizacija memorije

- Prvi dio predstavlja **fiksni kod**: Tabela vektora prekida, kod Operativnog sistema, kod aplikacije (kodovi svih procesa) i konstante.

Ovaj dio koda **se skoro uvek smešta u ROM memoriju.**

- Kod nekih procesora (mikrokontroleri i signal procesori) **ovaj deo memorije je i fizi ki odvojen od prostora za smeštaj podataka** pri emu se za pristup tim podacima koriste razli iti kontrolni signali.

Organizacija memorije

- **Drugi deo memorije** predstavlja RAM memorija koja je **zauzata od strane Operativnog sistema.**
- **Ovim delom memorije isključivo upravlja Operativni sistem.**
- **Korisniku se zabranjuje svaki direktan pristup** ovom memorijskom prostoru **jer on može biti fatalan po aplikaciju.**

Organizacija memorije

- U ovom delu memorije se smeštaju:
 - **Interne promenljive** Operativnog sistema;
 - **Kontrolni blokovi procesa** (PCB) aktivnih procesa;
 - **Sekundarne sistemske promenljive** - promenljive nastale pri kreiranju neke od kontrolnih struktura kojima Operativni sistem raspolaže ("mailbox", "queue", semafor, strukture za dodelu memorije...);
 - **Sistemske stekove** koji Operativni sistem koristi uvek kada se nalazi u okviru prekidne rutine;
 - **Stekovi za svaki od procesa**, kao i za multi proces.

Organizacija memorije

- **Tre i deo memorije** predstavlja slobodna memorija koja se **može dinami ki dodeljivati procesima i vra ati od strane procesa.**
- **To je u stvari deo kojim procesi mogu raspolagati i za koji oni konkurišu ukoliko je nedovoljan da zadovolji potrebe svih njih istovremeno**

Upravljanje memorijom

Upravljanje memorijom obuhvata:

- **vo** enje evidencije o svakoj eliji operativne memorije;
 - **odre** ivanje strategije dodeljivanja memorije;
 - **mehanizam dodele memorije**, to jest stavljanja memorije na raspolaganje,
 - mehanizam **osloba** anja memorije.
- Ovo je oblast u okviru koje su **razvijene brojne tehnike i mehanizmi**.
 - Razlog za to je u tome što **memorija, posle procesora, predstavlja najvažniji resurs ra unara** i od na ina njenog efikasnog koris enja **zavise i perfomanse** ra unarskog sistema.

Upravljanje memorijom

- **Naj eš e koris eni mehanizmi dodele (organizacije) memorije su:**
 - dodela (organizacija) memorije **u particijama** (fiksne ili promenljive dužine);
 - **strani na** dodela (organizacija) memorije;
 - **segmentna** dodela (organizacija) memorije;
 - **segmentno-strani na** dodela (organizacija) memorije;
 - **strani na virtuelna** dodela (organizacija) memorije;

Upravljanje memorijom

- **Jedan od najvećih problema** mehanizama za dodelu memorije jeste **fragmentacija** (usitnjavanje) memorije.
- **Procesi uzimaju i oslobađaju memoriju** u skladu sa svojim zahtevima i **redosled** ne može predvideti.
- Kao posledica toga vrlo brzo se dolazi do situacije da **postoji dovoljna količina slobodne memorije, ali da je ista izdeljena na veći broj delova (fragmenata)**, od kojih ni jedan ne poseduje dovoljnu veličinu da zadovolji novopristigli zahtev.
- Sa manje ili više efikasnosti ovaj problem se rešava **objedinjavanjem slobodne memorije** ili prihvatanjem kontrolisanog obima segmentacije u okviru definisane jedinice za dodelu memorije.

Upravljanje memorijom

- **Da bi se neka od ovih metoda koristila u operativnim sistemima za rad u realnom vremenu, ona mora zadovoljiti jednu vrlo važnu osobinu.**
- **Naime sistemi za rad u realnom vremenu moraju imati ograničeno vreme odziva na događaj tj. sve se mora odigrati u fiksnom vremenskom intervalu.**
- **Kod metoda koje vrše defragmentaciju spajanjem slobodne memorije ili deljenjem memorije i programa na strane i segmente vreme odziva se ne može unapred predvideti niti kontrolisati.**

Upravljanje memorijom

- Iz prethodnih razloga, za sisteme koji rade u realnom vremenu, naj češće korišćeni metod upravljanja memorijom je slede i:
 - Memorija koja je predviđena za dinamičku alokaciju deli se na regione fiksne veličine koji se nazivaju particijama.
 - Svaka particija se dalje deli na manje delove – blokove fiksne definisane dužine.
 - Blokovi se pri inicijalizaciji particije povezuju u listu slobodnih memorijskih blokova.
 - Kada neki od procesa uputi zahtev za memorijom, dodeljuje mu se blok iz liste slobodnih blokova tražene particije.
 - Ako ne postoji slobodan blok u datoj particiji, procesu se vraća odgovarajuća poruka. Kod nekih operativnih sistema, u ovakvoj situaciji sistemski poziv prouzrokuje suspenziju procesa.

Upravljanje memorijom

- Memorija koja procesu nije više potrebna **vraća se particiji**, pri čemu se vraćeni **blok uvršćuje u listu slobodnih blokova**.
- Ovim načinom je **izbegnuta eksterna fragmentacija** ali **ne i interna fragmentacija** u okviru samog memorijskog bloka.
- Time je plaćena cena kompromisu da se eksterna fragmentacija mora izbjeći i da **odziv na zahtev za memorijom bude definisanog trajanja**.

Upravljanje memorijom

- Za upravljanje memorijom se koriste **slede i sistemski pozivi:**
 - Sistemski poziv **za kreiranjem particije.**
 - Sistemski poziv **za dodeljivanje memorije.**
 - Sistemski poziv **za suspenziju procesa** ukoliko ne postoji slobodni memorijski prostor.
 - Sistemski poziv **za osloba anje memorije.**

Upravljanje memorijom

- **Sistemska poziv za kreiranjem particije.**
Ovaj poziv zauzima traženi memorijski prostor i deli ga na ulanane blokove slobodne memorije.
- **Sistemska poziv za dodeljivanje memorije**
Ovim pozivom operativni sistem procesu **vraća adresu alociranog bloka memorije**, pri čemu isti briše iz liste slobodnih blokova. Ukoliko ne postoji slobodan blok javlja se poruka greške procesu koji je uputio poziv.

Upravljanje memorijom

- **Sistemska poziv za suspenziju procesa.**
Neki sistemi imaju poziv za dodeljivanje memorije kojim se proces suspenduje ukoliko ne postoji slobodan memorijski blok.
- **Sistemska poziv za osloba anje memorije.**
Ovim pozivom proces vraća a operativnom sistemu adresu bloka koji mu više nije potreban.
Operativni sistem dati blok ubacuje u listu slobodnih memorijskih blokova.