

# Rukovanje Memorijom

UDŽBENIK, POGLAVLJE 12

# Zadatak sloja za rukovanje radnom memorijom

---

- Sloj za rukovanje radnom memorijom rukuje **fizičkom radnom memorijom**.
- Fizičkoj radnoj memoriji se pristupa posredstvom **fizičkog adresnog prostora** (koga koristi **operativni sistem**), ali i preko **logičkih adresnih prostora** (po jedan za svaki **korisnički proces**).
- Logički adresni prostori **izoluju procese** (**međusobno** i od **operativnog sistema**) tako što ograničavaju pristup procesa samo na deo fizičke radne memorije.

# Zadatak sloja za rukovanje radnom memorijom

---

- To se postiže **preslikavanjem** logičkog adresnog prostora svakog od procesa na samo **jedan deo** fizičkog adresnog prostora, koji odgovara delu fizičke memorije, dodeljene dotičnom procesu.
- Preslikavanje obavlja **MMU (Memory Management Unit)** tako što logičke adrese **pretvara (translira)** u odgovarajuće fizičke adrese.
- Funkcionisanje i organizacija **MMU** zavisi od karaktera logičkog adresnog prostora. Logički adresni prostor može biti:

# Zadatak sloja za rukovanje radnom memorijom

---

- 1) Kontinualan
- 2) Sastavljen od segmenata raznih veličina
- 3) Sastavljen od stranica iste veličine
- 4) Sastavljen od segmenata raznih veličina koji se sastoje od stranica iste veličine

# Kontinualni logički adresni prostor

---

- Kontinualni logički adresni prostor se sastoji od jednog niza **uzastopnih logičkih adresa**, koji počinje od logičke adrese **0**.
- Veličina kontinualnog logičkog adresnog prostora **nadmašuje** potrebe prosečnog procesa.
- Zato, da bi se mogao detektovati pokušaj izlaska procesa iz njegovog logičkog adresnog prostora, neophodno je poznavati **najvišu ispravnu** logičku adresu procesa.
- Ona se zove **granična adresa** procesa. Logičke adrese procesa **ne smeju biti veće** od njegove granične adrese.

# Kontinualni logički adresni prostor

---

- Za kontinualni logički adresni prostor se podrazumeva da se niz **uzastopnih logičkih** adresa **preslikava** u niz **uzastopnih fizičkih** adresa.
- Ova dva niza se razlikuju po tome što niz uzastopnih fizičkih adresa **ne počinje od 0** nego od neke adrese koja se zove **bazna adresa**.
- Znači, **dodavanjem bazne adrese logičkoj adresi** nastaje **fizička adresa**.
- U toku translacije logičke adrese u fizičku, **MMU poredi logičku i graničnu adresu**. Ako je logička adresa **veća**, MMU izaziva **izuzetak**, inače sabira **baznu adresu sa logičkom adresom** da bi dobio **fizičku adresu**.

# Segmentirani logički adresni prostor

---

- Segmentirani logički adresni prostor (**segmentation**) se sastoji od **više nizova** uzastopnih logičkih adresa (za svaki segment po **jedan niz**).
- Da bi se znalo kom segmentu pripada logička adresa, njeni **značajniji biti** sadrže adresu **njenog segmenta**, a **manje značajni biti** sadrže **unutrašnju adresu** u njenom segmentu.
- Pošto je svaki segment **kontinualan**, on ima svojstva **kontinualnog** logičkog adresnog prostora. Zato svaki segment karakterišu njegova **granična** i **bazna** adresa.

# Segmentirani logički adresni prostor

---

- U ovom slučaju za translaciju je potrebna **tabela segmenata**.
- Broj njenih elemenata određuje **najveći broj segmenata** u segmentiranom logičkom adresnom prostoru.
- Svaki element ove tabele sadrži **graničnu i baznu adresu odgovarajućeg segmenta**.
- Adresa segmenta** indeksira element tabele segmenata, a njegove **graničnu i baznu adresu** koristi **MMU** za translaciju **unutrašnje adrese segmenta** u **fizičku adresu**.



# Stranični logički adresni prostor

---

- **Stranični logički adresni prostor (paging)** se sastoji od **jednog niza** uzastopnih logičkih adresa (podeljenog u **stranice iste veličine**).
- Da bi se znalo **kojoj stranici** pripada logička adresa, njeni **značajniji biti** sadrže **adresu njene stranice**, a **manje značajni** biti sadrže **unutrašnju adresu** u **njenoj** stranici.
- Pošto je svaka stranica kontinualna, ona ima svojstva **kontinualnog** logičkog adresnog prostora.

# Stranični logički adresni prostor

---

- Razlika je da je veličina stranice **unapred poznata** i jednaka **stepenu broja dva**.
- Zato svaku stranicu karakteriše samo njena **bazna adresa** (graniča adresa nije potrebna, jer je unutrašnja adresa ograničena veličinom stranice).
- U ovom slučaju za translaciju je potrebna **tabela stranica**.
- Broj njenih elemenata jednak je **najvećem broju stranica** u straničnom logičkom adresnom prostoru.
- Svaki element ove tabele sadrži **baznu adresu odgovarajuće stranice**. Adresa stranice **indeksira** element tabele stranica, a njegovu baznu adresu koristi **MMU** za **translaciju logičke** adrese stranice u **fizičku** adresu.

# Odnos procesa i logičkog adresnog prostora

---

- Svaki proces mora da ima neophodne translacione podatke koji obuhvataju ili **graničnu i baznu adresu**, ili **tabelu segmenata**, ili **tabelu stranica**, ili **tabelu segmenata sa pripadnim tabelama stranica**.
- Translacione podatke pripremi operativni sistem, prilikom stvaranja procesa. Da bi mogao da obavlja **translaciju**, **MMU** mora da raspolaže sa odgovarajućim **translacionim podacima**.
- Translacioni podaci se menjaju prilikom **preključivanja**, pa utiču na trajanje **preključivanja**.
- **Veličina slike** procesa određuje veličinu njegovog **logičkog adresnog prostora**.

# Upotreba kontinualnog logičkog adresnog prostora

---

- Kontinualni logički adresni prostor se koristi kada je logički adresni prostor svakog procesa **manji od** raspoloživog fizičkog adresnog prostora.
- Da bi **translacija** logičkih adresa u fizičke bila moguća, neophodno je da se u lokacije fizičke radne memorije smesti slika procesa.
- U ovom slučaju, **translacija** logičkih adresa u fizičke adrese odgovara **dinamičkoj relokaciji** i olakšava **zamenu slika procesa (swapping)**.
- Pokušaj procesa da **izađe iz svog adresnog prostora** izaziva **izuzetak (SEGFAULT)** na koji reaguje operativni sistem, tako što **uništi** dotični proces.

# Upotreba segmentiranog logičkog adresnog prostora

---

- **Segmentacija** (segmentirani logički adresni prostor) se koristi kada je važno **racionalno korišćenje** fizičke radne memorije.
- I ovde se pretpostavlja da je logički adresni prostor procesa **manji od** raspoloživog fizičkog adresnog prostora, kao i da smeštanje slike procesa u lokacije fizičke radne memorije prethodi **translaciji** logičkih adresa u fizičke.
- U ovom slučaju, **segmenti** logičkog adresnog prostora procesa sadrže **delove** njegove slike.
- Na primer, postoje **segment za mašinske naredbe**, **segment za promenljive** i **segment za stek**.
- Ako istovremeno postoji **više** procesa, nastalih na osnovu **istog** programa, tada oni mogu da **dele iste mašinske naredbe**.

# Upotreba segmentiranog logičkog adresnog prostora

---

- U ovom slučaju, **tabele segmenata** takvih procesa sadrže **isti par granične i bazne** adrese **deljenog segmenta**.
- Ovaj par omogućuje **preslikavanje** unutrašnjih adresa segmenta naredbi raznih procesa na **iste** fizičke adrese lokacija fizičke radne memorije sa **deljenim** mašinskim naredbama.
- Izdvajanje **promenljivih i steka** procesa u **posebne segmente** je korisno i zbog mogućnosti naknadnog **proširenja** ovih segmenata (kada to postane potrebno u toku aktivnosti procesa).

# Upotreba segmentiranog logičkog adresnog prostora

---

- Prednost, **segmentacije** je da ona **ubrzava zamenu slika procesa (swapping)**, jer se segment naredbi ne mora **izbacivati**, ako je ranije već izbačen u masovnu memoriju, niti **ubacivati**, ako već postoji u fizičkoj radnoj memoriji.
- Jasno, rukovanje segmentima mora voditi posebnu evidenciju o segmentima, prisutnim u fizičkoj radnoj memoriji, da bi bilo moguće **otkriti** kada se isti segment može iskoristiti u slikama raznih procesa.

# Upotreba segmentiranog logičkog adresnog prostora

---

- Ovakva evidencija obuhvata jednoznačnu **oznaku segmenta**, podatak o **broju korisnika segmenta**, **dužinu segmenta**, odnosno, njegovu **graničnu** i **bazu** adresu.
- Radi očuvanja konzistentnosti evidencije segmenata, operacije za rukovanje ovom evidencijom moraju obezbediti **sinhronizaciju procesa**.



# Upotreba segmentiranog logičkog adresnog prostora

---

- Prethodno opisana (**osnovna**) segmentacija se razvija u **punu** segmentaciju, ako se dozvoli da svakom **potprogramu** ili **promenljivoj** odgovara **poseban** segment.
- To, na primer, dozvoljava **deljenje potprograma** između **raznih procesa**, ako se segment istog potprograma uključi u slike više procesa.
- U tom slučaju, stvaraju se uslovi za **dinamičko linkovanje (povezivanje)** potprograma za program.

# Upotreba segmentiranog logičkog adresnog prostora

---

- Ono se ne dešava u toku pravljenja izvršne datoteke, nego u trenutku **prvog poziva potprograma**, kada se u fizičkoj radnoj memoriji **pronalazi** njegov segment ili se **stvara** ako ne postoji.
- Dinamičko linkovanje doprinosi racionalnom korišćenju masovne memorije, jer omogućava da se **često korišćeni** potprogrami **ne multipliciraju** u raznim izvršnim datotekama.

# Upotreba segmentiranog logičkog adresnog prostora

---

- Puna segmentacija dozvoljava i **deljenje promenljivih** između raznih procesa, ako se segment iste promenljive uključi u slike više procesa.
- Jasno, ovakav način ostvarenja saradnje procesa zahteva njihovu **sinhronizaciju**, radi **očuvanja konzistentnosti** deljenih promenljivih.
- Za segmente sa deljenim promenljivima su važna i **prava pristupa** segmentu, jer nekim od procesa treba dozvoliti samo da **čitaju** deljenu promenljivu, a drugima treba dozvoliti i da **pišu** u deljenu promenljivu.

# Upotreba segmentiranog logičkog adresnog prostora

---

- Zato se elementi tabele segmenata proširuju **oznakom prava pristupa** segmentu.
- U pogledu prava pristupa, segmenti se **ne razlikuju** od datoteka.
- Prema tome, prava pristupa segmentu obuhvataju pravo **čitanja**, pravo **pisanja** i pravo **izvršavanja** segmenta.
- Prva dva prava se primenjuju na segmente, koji sadrže **promenljive**, a treće pravo se primenjuje na segmente sa **mašinskim naredbama** programa ili potprograma.

# Upotreba segmentiranog logičkog adresnog prostora

---

- Na pokušaj **narušavanja prava pristupa** segmentu reaguje MMU, generisanjem **prekida (izuzetka)**, što dovodi do uništenja aktivnog procesa (**SEGFault**).
- Rukovanje segmentima se obavlja posredstvom sistemskih programa, kao što su **kompajler** ili **linker**.

# Upotreba straničnog logičkog adresnog prostora

---

- **Stranični** logički adresni prostor se koristi kada je logički adresni prostor tipičnog procesa **veći** od raspoloživog fizičkog adresnog prostora, pa slika procesa **ne može** da stane u fizičku radnu memoriju.
- Stranični logički adresni prostor se naziva i **virtuelni adresni prostor**.
- On pripada **virtuelnoj memoriji** i sadrži **virtuelne adrese**.

# Upotreba straničnog logičkog adresnog prostora

---

- Stranice **virtuelnog** adresnog prostora sa slikom procesa se nalaze na **masovnoj memoriji**.
- Postoji posebna **evidencija** o tome gde se one tačno nalaze na masovnoj memoriji.
- Pošto je za izvršavanje mašinske naredbe neophodno da u fizičkoj radnoj memoriji budu samo **bajti njenog mašinskog formata**, kao i **bajti njenih operanada**, u fizičkoj radnoj memoriji moraju da se nalaze samo **kopije virtuelnih stranica** koje sadrže **pomenute bajte**.

# Upotreba straničnog logičkog adresnog prostora

---

- Podrazumeva se da se kopije neophodnih virtuelnih stranica, kada zatreba, **automatski prebacuju** iz masovne memorije u fizičku radnu memoriju i obrnuto.
- Do prebacivanja u obrnutom smeru dolazi, kada je potrebno **osloboditi** lokacije fizičke radne memorije sa **kopijama** u **međuvremenu izmenjenih** virtuelnih stranica.
- Ovo prebacivanje se obavlja, da bi se **oslobodila fizička radna memorija** i, istovremeno, obezbedilo da masovna memorija uvek sadrži **ažurnu sliku procesa**.
- Da bi se znalo koja kopija virtuelne stranice je **izmenjena**, neophodno je **automatski registrovati svaku izmenu svake od kopija** virtuelnih stranica.



# Upotreba straničnog logičkog adresnog prostora

---

- Fizička radna memorija sadrži **kopije** pojedinih **virtuelnih** stranica, pa je prirodno da i ona bude **izdeljena** u fizičke stranice u kojima se nalaze **kopije** virtuelnih stranica.
- To znači da se fizička adresa sastoji od **adrese fizičke stranice** (u **značajnijim bitima** fizičke adrese) i od **unutrašnje adrese** (u **manje značajnim bitima** fizičke adrese).
- Pošto su virtuelne i fizičke stranice **iste veličine**, unutrašnja adresa (manje značajni biti) virtuelne adrese se **poklapa** sa unutrašnjom adresom (manje značajnim bitima) fizičke adrese.

# Upotreba straničnog logičkog adresnog prostora

---

- Zato je za **translaciju** virtuelne adrese u fizičku potrebno samo **zameniti** adresu **virtuelne stranice** adresom **fizičke stranice**.
- Zbog toga se u elementu **tabele stranica** (koga indeksira adresa **virtuelne stranice**) kao **bazna adresa** nalazi **adresa fizičke** stranice (koja sadrži kopiju dotične **virtuelne stranice**).
- Ako virtuelna stranica nije kopirana u neku fizičku stranicu, tada translacija njenih virtuelnih adresa u fizičke nije moguća.

# Upotreba straničnog logičkog adresnog prostora

---

- U tom slučaju, to mora biti registrovano u odgovarajućem **elementu tabele stranica**. Takvu ulogu ima **bit prisustva**.
- Uz bit prisustva, elementi tabele stranica sadrže i **bit referenciranja** (koji pokazuje da li je bilo **pristupanja** kopiji virtuelne stranice) i **bit izmene** (koji pokazuje da li je bilo **izmena** kopije virtuelne stranice).
- Poslednja dva bita se **automatski** postavljaju.

# Upotreba straničnog logičkog adresnog prostora

---

- Kada pokušaj **translacije** virtuelne adrese bude **neuspešan** (jer **bit prisustva** ukaže da se kopija odgovarajuće virtuelne stranice **ne nalazi** u fizičkoj radnoj memoriji), MMU izaziva **stranični prekid (page fault)**.
- U obradi straničnog prekida se prebaci **kopija** potrebne **virtuelne stanice** u neku **fizičku stranicu**.
- Adresa ove fizičke stranice postaje **bazna adresa** i smesti se u odgovarajući element **tabele stranica**.
- U ovom elementu se istovremeno postavi **bit prisustva**, a **očiste** se **bit referenciranja** i **bit izmene**.
- Nakon toga, **ponavlja se** neuspešna translacija virtuelne adrese.

# Upotreba straničnog logičkog adresnog prostora

---

- Važno je uočiti da se kopije virtuelnih stranica prebacuju na **zahtev (demand paging)**, a ne **unapred (prepaging)**, jer u opštem slučaju ne postoji način da se **predvidi** redosled korišćenja virtuelnih stranica.
- Praktična upotrebljivost virtuelne memorije se temelji na svojstvu **lokalnosti** izvršavanja programa.
- Zahvaljujući ovome svojstvu, za izvršavanje programa je dovoljno da u fizičkoj radnoj memoriji uvek bude **samo deo** programa.

## Upotreba stranično segmentiranog logičkog adresnog prostora

---

- **Stranična segmentacija** (stranično segmentirani logički adresni prostor) se koristi kada su segmenti potrebni radi **racionalnog** korišćenja fizičke radne memorije, a njihova veličina **nadmašuje** veličinu raspoložive fizičke radne memorije.
- U tom slučaju, svaki **segment** uvodi sopstveni **virtuelni** adresni prostor.
- Zahvaljujući tome, stranice **virtuelnog** adresnog prostora segmenta se nalaze u **masovnoj** memoriji, a u **fizičkim** stranicama se nalaze samo **kopije neophodnih virtuelnih** stranica segmenta.

## Upotreba stranično segmentiranog logičkog adresnog prostora

---

- Prednost stranične segmentacije je da ona omogućava **dinamičko proširenje segmenata** (dodavanjem novih stranica), što je važno za **segmente promenljivih i steka**.
- Stranična segmentacija može otvorenim datotekama dodeljivati **posebne segmente** i na taj način ponuditi koncept **memorijski preslikane datoteke (memory mapped file)**.
- Pristup ovakvoj datoteci ne zahteva sistemske operacije za **čitanje, pisanje ili pozicioniranje**, jer se **direktno** pristupa **lokacijama sa odgovarajućim sadržajem** datoteke.

## Upotreba stranično segmentiranog logičkog adresnog prostora

---

- **Mana** koncepta memorijski preslikane datoteke je da se veličina datoteke izražava **celim brojem stranica**, jer nema načina da operativni sistem odredi koliko je **popunjeno bajta** iz **poslednje stranice**.
- Takođe, problem je i što virtuelni adresni prostor segmenta može biti **suviše mali** za pojedine datoteke.



## Zadaci sloja za rukovanje fizičkom radnom memorijom

---

- Zadatak sloja za rukovanje fizičkom radnom memorijom je da omogući **zauzimanje zona susednih lokacija** (sa uzastopnim adresama) slobodne fizičke radne memorije, kao i da omogući **oslobađanje prethodno zauzetih zona** fizičke radne memorije.
- Radi toga ovaj sloj nudi operacije **zauzimanja** i **oslobađanja** (fizičke radne memorije).

## Zadaci sloja za rukovanje fizičkom radnom memorijom

---

- Argument poziva operacije zauzimanja je **dužina zauzimane zone** (broj njenih lokacija), a **povratna** vrednost ovog poziva je **adresa zauzete zone** (adresa prve od njenih lokacija), ili indikacija da je operacija zauzimanja završena **neuspešno**.
- Argumenti poziva operacije oslobađanja su **adresa oslobađane zone** (adresa prve od njenih lokacija) i njena **dužina** (broj njenih lokacija).

## Zadaci sloja za rukovanje fizičkom radnom memorijom

---

- Uspešno obavljanje zadatka sloja za rukovanje fizičkom radnom memorijom se temelji na vođenju **evidencije** o slobodnoj fizičkoj radnoj memoriji.
- Kada podržava virtuelnu memoriju, ovaj sloj mora svakom procesu dodeliti dovoljan broj fizičkih stranica za kopije potrebnih virtuelnih stranica.

# Raspodela fizičke radne memorije

---

- Fizička radna memorija se deli na lokacije koje stalno zauzima **operativni sistem** i na **preostale slobodne lokacije** koje su na raspolaganju za **stvaranje procesa** i za druge potrebe.
- **Operativni sistem** obično zauzima lokacije **sa početka** i, eventualno, **sa kraja** fizičkog adresnog prostora, a **između njih** se nalaze lokacije **slobodne fizičke radne memorije**.
- Na **početku** fizičkog adresnog prostora su, najčešće, lokacije, namenjene za **tabelu prekida**.

# Raspodela fizičke radne memorije

---

- Iza njih slede lokacije sa **naredbama** i **promenljivim** operativnog sistema (koje obuhvataju i prostor za smeštanje **deskriptora** i **sistemskih stekova** procesa).
- Na kraju fizičkog adresnog prostora su, najčešće, lokacije, koje odgovaraju **registrima kontrolera**.

# Raspodela fizičke radne memorije

---

- U slučaju da procesor podržava **virtuelnu memoriju**, praksa je da se samo **donja polovina** (sa nižim adresama) virtuelnog adresnog prostora stavi na raspolaganje **svakom procesu**, a da **gornja polovina** (sa višim adresama) virtuelnog adresnog prostora bude rezervisana za **operativni sistem**.
- **Gornja polovina** virtuelnog adresnog prostora se zato može nazvati **sistemska virtuelni adresni prostor**, a **donja polovina** virtuelnog adresnog prostora se može nazvati **korisnički virtuelni adresni prostor**.

# Raspodela fizičke radne memorije

---

- Za virtuelne adrese iz **prve polovine** sistemskog virtuelnog adresnog prostora se **ne vrši translacija** (na primer, tako što se deo manje značajnih bita virtuelne adrese koristi kao fizička adresa) i podrazumeva se da se tu nalazi **rezidentni deo operativnog sistema**, koji je **stalno prisutan** u fizičkoj radnoj memoriji.
- Virtuelne adrese iz **druge polovine** sistemskog virtuelnog adresnog prostora **se transliraju** na uobičajeni način i podrazumeva se da se one odnose na **nerezidentni deo operativnog sistema**, koji se po potrebi **prebacuje** u fizičku radnu memoriju.

# Evidencija slobodne fizičke radne memorije

---

- Sloj za rukovanje fizičkom radnom memorijom obavezno vodi evidenciju o **slobodnoj** fizičkoj radnoj memoriji.
- Za potrebe kontinualnog ili segmentiranog logičkog adresnog prostora ova evidencija može da bude u obliku **niza bita (bit map)**, u kome svaki bit odgovara grupi **susednih lokacija**. Podrazumeva se da je broj lokacija u ovakvoj grupi unapred zadan.



# Evidencija slobodne fizičke radne memorije

---

- On ujedno predstavlja jedinicu u kojoj se izražava dužina **zaузimane** i **oslobađane** zone fizičke radne memorije.
- Ako je grupa lokacija **slobodna**, njoj odgovarajući bit sadrži **1**. Inače, on sadrži **0**.
- Mana ovakve evidencije je da su i operacija **zaузimanja** i operacija **oslobađanja dugotrajne**, jer prva **pretražuje evidenciju**, radi pronalaženja **dovoljno dugačkog** niza **jedinica**, a druga **postavlja** takav niz jedinica u evidenciju.

# Evidencija slobodne fizičke radne memorije

---

- Zato se češće evidencija slobodne fizičke radne memorije pravi u obliku **liste slobodnih odsečaka** fizičke radne memorije.
- Na početku rada operativnog sistema ovakva lista sadrži **jedan odsečak**, koji obuhvata **celu fizičku slobodnu radnu memoriju**.
- Ovakav odsečak se **drobi** u **više kraćih odsečaka**, kao rezultat **višestrukog stvaranja i uništavanja procesa** u slučajnom redosledu.

# Evidencija slobodne fizičke radne memorije

---

- Novonastali kraći odsecci se nalaze na mestu slika **uništenih** procesa, a između njih su slike **postojećih** procesa.
- Na početku svakog odsečka su **njegova dužina** i **adresa narednog** odsečka.
- Broj lokacija, potrebnih za smeštanje **dužine** dotičnog odsečka i **adrese** narednog odsečka, određuje **najmanju dužinu odsečka** i može da predstavlja **jedinicu** u kojoj se izražavaju dužine odsečaka (odnosno, dužine zauzimanih i oslobađanih zona fizičke radne memorije).

# Evidencija slobodne fizičke radne memorije

---

- Odsecci su uređeni u **rastućem** redosledu adresa njihovih **početnih lokacija**. To, prilikom **oslobađanja** zone fizičke radne memorije, **olakšava operaciji oslobađanja**:
  - da **pronađe dva susedna odsečka**, koji mogu da se **spoje u jedan**, kada se **između njih ubaci oslobađana zona**, ili
  - da **pronađe odsečak**, kome može da se **doda (spređa ili straga) oslobađana zona** ili
  - da **pronađe mesto u listi** u koje će oslobađana zona biti uključena kao **poseban odsečak**.

# Evidencija slobodne fizičke radne memorije

---

- Listu slobodnih odsečaka **pretražuje** i operacija **zauzimanja**, radi pronalaženja **dovoljno dugačkog** odsečka.
- Pri tome se zauzima samo **deo odsečka**, koji je jednak **zauzimanoj zoni**, dok preostali deo odsečka **ostaje** u listi kao **novi** odsečak.
- Na ovaj način se odsecci dalje **usitnjavaju**. To dovodi do **eksterne fragmentacije**.

# Evidencija slobodne fizičke radne memorije

---

- Ona posredno uzrokuje **neupotrebljivost** odsečaka, jer **onemogućuje** zauzimanje zone fizičke radne memorije, čija dužina je **veća od dužine svakog od** postojećih odsečaka, bez obzira na činjenicu da je **suma dužina postojećih odsečaka veća od dužine** zauzimate zone.
- Iskustvo pokazuje da se **eksterna fragmentacija povećava**, ako se, umesto traženja **prvog dovoljno dugačkog odsečka (first fit)**, pokušava naći **najmanji dovoljno dugačak odsečak (best fit)**, ili **najveći dovoljno dugačak odsečak (worst fit)**.

# Evidencija slobodne fizičke radne memorije

---

- Poboljšanje **ne nudi** ni ideja da lista odsečaka bude **ciklična** i da se pretražuje **ne od početka, nego od tačke** u kojoj je zaustavljeno **poslednje pretraživanje (next fit)**.
- Ideja da dužina zauzimanih zona bude uvek jednaka **stepenu broja 2 (quick fit)** i da postoji posebna lista odsečaka **za svaku od mogućih dužina** **takođe ima manu**, jer, pored **eksterne**, uvode i **internu** fragmentaciju, pošto se na ovaj način u proseku zauzimaju **duže zone od stvarno potrebnih**, čime nastaje **neupotrebljiva radna memorija**.

# Evidencija slobodne fizičke radne memorije

---

- Problem eksterne fragmentacije se može rešiti **sabijanjem (compaction)** slika procesa, čime se sve slike procesa pomeraju na **jedan kraj fizičke radne memorije**, tako da na **drugom kraju** bude **slobodna fizička radna memorija**.
- U toku sabijanja, moraju se menjati **bazne adrese (segmenata)** pojedinih procesa.
- Mana sabijanja je njegova **dugotrajnost (sporost)**.



# Evidencija slobodne fizičke radne memorije

---

- Za potrebe virtuelnog adresnog prostora, evidencija slobodne fizičke radne memorije može da bude u obliku **niza bita**, u kome **svaki bit** odgovara **slobodnoj fizičkoj** stranici.
- Alternativa je da se slobodne fizičke stranice vežu u **listu**.
- Međutim, atraktivna je i evidencija o **slobodnim odsečcima** veličine jednake **multiplu fizičke stranice (quick fit, buddy system)**, jer se u virtuelnom adresnom prostoru uvek zauzima **celi broj stranica**.

# Evidencija slobodne fizičke radne memorije

---

- Važno je uočiti da rukovanje evidencijom sistemske slobodne radne memorije zahteva **sinhronizaciju procesa**, u toku čije aktivnosti se istovremeno obavljaju operacije **zauzimanja** i **oslobađanja**.
- Sinhronizacija treba da obezbedi **međusobnu isključivost** obavljanja ovih operacija, radi očuvanja **konzistentnosti** pomenute evidencije.

# Evidencija slobodne fizičke radne memorije

---

- Za virtuelnu memoriju je važno pitanje **dužine stranice**. Za **dugačke stranice** postaje izražen problem **interne fragmentacije**, jer sve stranice nisu uvek **potpuno iskorišćene**, pa se u njima javljaju **neupotrebljive lokacije**.
- Za **kratke stranice** postaje izražen problem **veličine tabele stranica**, jer tada virtuelni adresni prostor ima više stranica, pa zato i tabela stranica ima **više elemenata**.
- Praksa je veličinu stranice smestila između **512** i **8192** bajta.

# Dodela fizičkih stranica procesima

---

- Za aktivnost procesa je potrebno da procesoru na raspolaganju budu **kopije svih virtuelnih stranica** koje su neophodne za izvršavanje pojedinih mašinskih naredbi.
- Za čuvanje kopija tih stranica procesu mora biti dodeljeno **dovoljno fizičkih stranica** koje obrazuju **minimalan skup**.

# Dodela fizičkih stranica procesima

---

- Na primer, za procesor, čije naredbe imaju najviše **dva operanda**, minimalni skup sadrži **šest fizičkih stranica**, jer se, u **ekstremnom** slučaju, i **bajti mašinske naredbe**, kao i **bajti oba njena operanda**, mogu nalaziti u **različitim** susednim fizičkim stranicama.
- Pošto se naredba može izvršiti samo kada su u fizičkoj radnoj memoriji prisutni **svi bajti njenog mašinskog formata** i **svi bajti njenih operanada**, prethodno pomenuti ekstremni slučaj uslovljava da je pridruživanje **minimalnog** skupa procesu preduslov bilo kakve njegove aktivnosti.

# Dodela fizičkih stranica procesima

---

- Kada se, u toku aktivnosti procesa, desi **stranični prekid**, koji zahteva prebacivanje **kopije nove virtuelne stranice** u radnu memoriju, pre zahtevanog prebacivanja neophodno je razrešiti dilemu da li **uvećati** skup fizičkih stranica procesa novom fizičkom stranicom i u nju smestiti kopiju nove virtuelne stranice, ili u **postojećem** skupu fizičkih stranica procesa zameniti sadržaj neke od njih kopijom nove virtuelne stranice.

# Dodela fizičkih stranica procesima

---

- **Uvećanje skupa fizičkih stranica** procesa ima smisla samo ako to dovodi do  **smanjivanja učestanosti straničnih prekida** (njihovog prosečnog broja u jedinici vremena).
- Znači, kada je, u toku aktivnosti procesa, **učestanost straničnih prekida iznad** neke (iskustveno određene) gornje granice, tada ima smisla **uvećanje skupa fizičkih stranica** procesa, da bi se učestanost straničnih prekida svela na prihvatljiv nivo.

# Dodela fizičkih stranica procesima

---

- To je važno, jer **obrada** svakog **prekida** troši **procesorsko vreme**, pa **veliki broj obrada straničnih prekida** može u potpunosti da angažuje procesor i da tako vrlo **uspori**, ili potpuno **spreči** njegovu bilo kakvu korisnu **aktivnost (trashing)**.
- Međutim, ako je učestanost straničnih prekida **ispod** neke (iskustveno određene) **donje** granice, tada ima smisla **smanjenje skupa** fizičkih stranica procesa, jer i sa **manjim skupom** fizičkih stranica učestanost straničnih prekida ostaje u prihvatljivom rasponu.



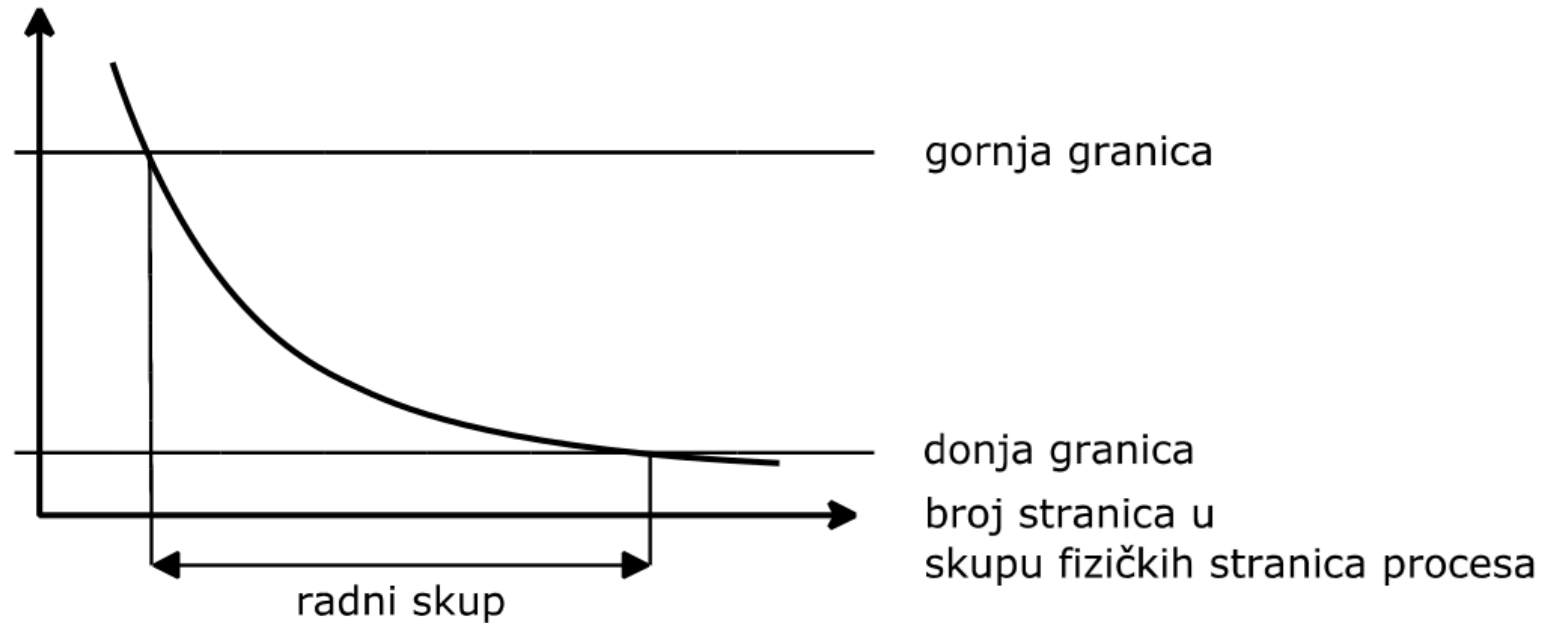
# Dodela fizičkih stranica procesima

---

- U opštem slučaju učestanost **straničnih prekida** ne pada na **nulu**, ako sve kopije virtuelnih stranica procesa ne mogu stati u radnu memoriju.
- **Smanjenje skupa** fizičkih stranica procesa je važno, jer se tako omogućuje neophodni **rast** skupova stranica **drugih** procesa.
- U slučaju da je učestanost straničnih prekida **između** pomenute dve granice, tada nema potrebe za izmenom broja fizičkih stranica u skupu fizičkih stranica aktivnog procesa.
- U ovom slučaju skup fizičkih stranica (odnosno, njima odgovarajući skup virtuelnih stranica) obrazuje **radni skup (working set)**.

# Dodela fizičkih stranica procesima

---



Odnos učestanosti straničnih prekida i broja stranica u skupu fizičkih stranica procesa

# Dodela fizičkih stranica procesima

---

- Radni skup procesa nije **statičan**.
- U proseku, on se **sporo menja** u toku aktivnosti procesa (iako su, povremeno, moguće značajne kratkotrajne varijacije radnog skupa).
- Važno je uočiti da se u toku aktivnosti procesa obavezno javlja **trashing**, kada njegov radni skup **ne može da stane** u radnu memoriju.
- U ovom slučaju pomaže **izbacivanje (swapping) procesa**, dok se ne oslobodi dovoljan broj fizičkih stranica.
- Ovaj pristup ima smisla samo ako je radni skup procesa **manji** od ukupne slobodne fizičke radne memorije.

# Dodela fizičkih stranica procesima

---

- **Stepen multiprogramiranja** kod virtuelne memorije zavisi od **broja radnih skupova**, koji se istovremeno mogu smestiti u raspoloživu fizičku radnu memoriju.
- Za uspeh koncepta virtuelne memorije važno je da se stalno prate **radni skupovi istovremeno postojećih procesa** i da se povremeno **izbacuju procesi**, čim fizička radna memorija postane **pretesna** za sve radne skupove (**load control**).
- Na ovaj način se **oslobađaju fizičke stranice** za preostale procese, neophodne za smeštanje njihovih radnih skupova.
- Prilikom kasnijeg ubacivanja procesa, uputno je ubacivati kopije svih virtuelnih stranica, koje obrazuju njegov radni skup.

# Oslobađanje fizičkih stranica procesa

---

- **Pad učestanosti straničnih prekida** ispod donje granice ukazuje na mogućnost smanjenja radnog skupa.
- U ovoj situaciji je potrebno odlučiti koju **virtuelnu stranicu izbaciti** iz radnog skupa, odnosno **osloboditi**.
- Za oslobađanje kao kandidat se nameće virtuelna stranica, koja **neće biti referencirana** do kraja aktivnosti procesa, ili će biti **referencirana** iza svih ostalih **virtuelnih stranica** iz radnog skupa (pod referenciranjem se podrazumeva pristup bilo kojoj lokaciji stranice, radi preuzimanja ili izmene njenog sadržaja).

# Oslobađanje fizičkih stranica procesa

---

- **Povećanje učestanosti** straničnih prekida preko gornje granice ukazuje na **potrebu proširenja radnog skupa**.
- U ovom slučaju, potrebno je procesu pridružiti **novu** fizičku stranicu, da bi se u nju smestila **kopija** potrebne virtuelne stranice.
- Ako nema slobodnih fizičkih stranica, tada se oslobađa, kada je to moguće, fizička stranica, koja je pridružena nekom **drugom procesu**.
- Time se radni skup ovog drugog procesa **smanjuje**. U oslobođenoj fizičkoj stranici zatečenu kopiju **zamenjuje (replacement)** kopija potrebne virtuelne stranice.

# Oslobađanje fizičkih stranica procesa

---

- Do **zamene kopija** virtuelnih stranica dolazi i kada se javi potreba za **ubacivanjem kopije** nove stranice, a učestanost straničnih prekida je između **gornje** i **donje** granice, pa se radni skup aktivnog procesa **ne proširuje**, nego se **oslobađa** jedna od fizičkih stranica iz njegovog **radnog skupa**.
- U oba prethodna slučaja kandidat za zamenu je fizička stranica, koja sadrži kopiju virtuelne stranice sa **najstarijom referencom**.

# Oslobađanje fizičkih stranica procesa

---

- I u slučaju **pada učestanosti** straničnih prekida **ispod donje granice** i u slučaju **povećanja učestanosti** straničnih prekida **iznad gornje granice**, kao i u slučaju kada je učestanost straničnih prekida **između gornje i donje granice**, javlja se potreba za **oslobađanjem** fizičkih stranica.
- Izbor fizičke stranice za oslobađanje se može vršiti **na razne načine**, po raznim algoritmima.
- Ovakvi algoritmi se nazivaju algoritmi **zamene stranica** (**page replacement algorithms**), jer se zatečeni sadržaj oslobađane fizičke stranice **zamenjuje** novim sadržajem.



# Oslobađanje fizičkih stranica procesa

---

- Oslobađanje fizičkih stranica može biti zasnovano samo na upotrebi **bita referenciranja** i **bita izmene**.
- U ovom pristupu se pronalazi fizička stranica koja **nije korišćena** u **nekom prethodnom periodu (Not Recently Used - NRU)** tako što se posmatra **dvobitni broj**, čiji **značajniji** bit odgovara bitu **referenciranja**, a **manje značajan** bit odgovara bitu **izmene (MMU ih postavlja)** .

# Oslobađanje fizičkih stranica procesa

---

- Takođe se podrazumeva da se **oba bita čiste** i prilikom **oslobađanja**, odnosno prilikom **zauzimanja** fizičke stranice (prilikom zamene njenog sadržaja).
- Prema tome, pomenuti dvobitni broj sadrži **00**, kada kopija virtuelne stranice **nije korišćena** nakon nekog od poslednjih vremenskih prekida (znači, **niti referencirana niti izmenjena**).
- Pomenuti dvobitni broj sadrži **01**, kada kopija virtuelne stranice **nije referencirana** nakon nekog od poslednjih vremenskih prekida (**ali je prethodno izmenjena**).

# Oslobađanje fizičkih stranica procesa

---

- Ovaj broj sadrži **10**, kada kopija virtuelne stranice nije **izmenjena**, ali je **referencirana** nakon poslednjeg vremenskog prekida.
- Pomenuti dvobitni broj sadrži **11**, kada je kopija virtuelne stranice **izmenjena** i, uz to, **referencirana** nakon poslednjeg vremenskog prekida.
- Kandidati za zamenu su stranice, čiji dvobitni broj je **najmanji**.
- Prethodno opisani pristup oslobađanja fizičkih stranica je **efikasan**, ali **nedovoljno precizno procenjuje** koja fizička stranica će **biti ubrzo referencirana**.

# Oslobađanje fizičkih stranica procesa

---

- U opštem slučaju **nema načina** da se **precizno ustanovi** da li će i kada neka virtuelna stranica biti **referencirana**.
- Ipak, zahvaljujući **lokalnosti referenciranja**, odnosno zapažanju da se pristupi lokacijama sa **bliskim adresama** dešavaju u **bliskim trenucima**, moguće je sa priličnom pouzdanošću zaključivati o **referenciranju stranica** u **neposrednoj budućnosti** na osnovu njihovog referenciranja u **neposrednoj prošlosti**.
- Prema tome, **najmanju verovatnoću** da bude referencirana u **neposrednoj budućnosti** ima stranica, čije poslednje referenciranje je **najstarije**, odnosno prethodi referenciranju svih ostalih stranica iz radnog skupa.

# Oslobađanje fizičkih stranica procesa

---

- Za pronalaženje **najmanje korišćene** fizičke stranice (**Least Recently Used - LRU**), odnosno stranice sa **najstarijom referencom**, neophodno je registrovanje starosti referenciranja.
- To je moguće, ako postoji **brojač**, koji se automatski **uvećava za jedan** nakon izvršavanja svake naredbe.
- Ako se njegova zatečena vrednost automatski **pridružuje virtuelnoj stranici** prilikom njenog svakog referenciranja, tada je stranici sa **najstarijom referencom** pridružena **najmanja vrednost** ovog brojača.

# Oslobađanje fizičkih stranica procesa

---

- Opisani pristup ima samo teoretsko značenje, jer se oslanja na **hardver**, koji u opštem slučaju nije raspoloživ.
- Međutim, moguće je i **softverski simulirati** pronalaženje **najmanje korišćene** fizičke stranice (**Not Frequently Used – NFU/aging**).
- Ovakva simulacija se zasniva na korišćenju **bita referenciranja** svake virtuelne stranice i na uvođenju **polja starosti referenci** u elemente tabele stranica.
- Ovo polje sadrži **n** bita, po **jedan bit** za svaku od vrednosti bita referenciranja u poslednjih **n trenutaka**.

# Oslobađanje fizičkih stranica procesa

---

- Polje starosti referenci se periodično **ažurira**, tako što se iz njega periodično **izbacuje najstariji bit** referenciranja.
- To je **zadatak obrađivača vremenskog prekida** koji:
  - pomera **u desno za jedan bit polje starosti referenci** svake od virtuelnih stranica iz radnog skupa aktivnog procesa
  - **dodaje s leva, u upražnjenu bitnu poziciju ovog polja, zatečeni sadržaj bita referenciranja dotične virtuelne stranice**
  - **zatim očisti bit referenciranja.**
- Na ovaj način polje starosti referenci sadrži **najmanju vrednost** za virtuelnu stranicu, koja je **najranije referencirana** u prethodnih **n trenutaka**.

# Oslobađanje fizičkih stranica procesa

---

- Polje starosti referenci se može iskoristiti i za određivanje **radnog skupa**.
- Kriterijum može biti postavljenost nekog od **k najznačajnijih bita** iz polja starosti referenci.
- Po tom kriterijumu radnom skupu pripadaju sve virtuelne stranice, koje imaju bar **jedan bit postavljen** u **najznačajnijih k bita** svog polja starosti referenci.



# Oslobađanje fizičkih stranica procesa

---

- Prethodno opisani pristupi oslobađanja fizičkih stranica (**NRU, LRU, NFU**) obezbeđuju **smanjenje učestanosti** straničnih prekida nakon **povećanja broja fizičkih stranica procesa**.
- Znači, obavljanje liste istih zahteva za pristupe virtuelnim stranicama dovodi do pojave **manje straničnih prekida** nakon **povećanja broja fizičkih stranica procesa**, nego pre toga.

# Oslobađanje fizičkih stranica procesa

---

- Ovo je važno istaći, jer svi pristupi oslobađanja fizičkih stranica ne dovode obavezno do smanjenja učestanosti straničnih prekida nakon povećanja broja fizičkih stranica procesa.
- To je karakteristično, na primer, za pristup, kod koga se oslobađa fizička stranica sa **najstarijim sadržajem (First In First Out - FIFO)**, bez obzira da li je ona skoro referencirana.
- Ovaj pristup ima tendenciju da ne oslobađa fizičke stranice, čiji sadržaj je svežiji, čak i ako one neće biti uskoro referencirane.

# Oslobađanje fizičkih stranica procesa

---

- Međutim, mana poslednje pomenutog pristupa se otklanja, ako se on **modifikuje**, tako da se za oslobađanje prvo traži fizička stranica sa **najstarijim** nereferenciranim sadržajem, a ako su sadržaji svih fizičkih stranica **referencirani**, tek tada se oslobađa fizička stranica sa **najstarijim sadržajem** (**second chance** i **clock** pristupi).
- U sva tri poslednje pomenuta pristupa (**FIFO**, **second chance** i **clock**) fizičke stranice procesa se uvezuju u **listu**.

# Oslobađanje fizičkih stranica procesa

---

- Pri tome **položaj** u listi ukazuje na **starost** sadržaja fizičke stranice.
- U **clock** algoritmu zamene ovakva lista je **kružna**, a poseban pokazivač, kao **kazaljka na satu**, pokazuje na stranicu sa **najstarijim sadržajem**.
- Postoji i varijanta **clock** pristupa (**wsclock**) u kome se uz svaku fizičku stranicu iz liste čuva i podatak o **trenutku poslednjeg referenciranja**.
- Za sve fizičke stranice, za koje ovaj trenutak **ispada** iz unapred određenog vremenskog intervala (gledajući u prošlost), se smatra da **ne spadaju u radni skup**, pa su one **kandidati za zamenu**.

# Oslobađanje fizičkih stranica procesa

---

- Pristup **NFU/aging** i pristup **wsclock** imaju najveću praktičnu važnost.
- Virtuelna memorija pokazuje najbolje rezultate, kada uvek ima slobodnih fizičkih stranica.
- To se može postići, ako se uvede poseban **stranični sistemski proces**, koji se periodično aktivira, da bi **oslobodio** izvestan broj fizičkih stranica.
- On, pri tome, odabira fizičke stranice za oslobađanje po nekom od **prethodno opisanih algoritama** zamene.
- Zadatak straničnog sistemskog procesa je i da u masovnu memoriju prebacuje **izmenjene kopije** virtuelnih stranica i da tako čuva **ažurnost** masovne memorije.

# Oslobađanje fizičkih stranica procesa

---

- Opisani pristupi oslobađanja fizičkih stranica pokazuju da se u praksi **ne prati** učestanost straničnih prekida.
- Umesto toga, broj fizičkih stranica procesa varira između **minimalnog skupa** i iskustveno određenog **maksimalnog skupa**.
- Pri tome stranični prekidi izazivaju **povećanje broja** fizičkih stranica procesa do maksimalne veličine, a stranični sistemski proces se brine o **njegovom smanjivanju**.

# Implementacija upravljanja virtuelnom memorijom

---

- Sloj za rukovanje virtuelnom memorijom podržava operacije **zauzimanja** i **oslobađanja**, a oslanja se na stranični sistemski proces i obrađivače **vremenskog** i **straničnog** prekida.
- Obradivač **straničnog prekida** se aktivira, kada je referencirana **virtuelna stranica**, čija kopija **nije prisutna** u fizičkoj radnoj memoriji, odnosno, u nekoj od njenih fizičkih stranica.
- Ako je, greškom, referencirana virtuelna stranica, koja uopšte **ne postoji** u slici procesa, obrađivač straničnog prekida **završava** aktivnost prekinutog procesa (uz odgovarajuću poruku).

# Implementacija upravljanja virtuelnom memorijom

---

- Inače, ovaj obrađivač odabira **slobodnu** fizičku stranicu i prema njoj usmerava prenos kopije potrebne virtuelne stranice sa masovne memorije.
- Kada se taj prenos završi, obrađivač straničnog prekida **ažurira** polja odgovarajućeg elementa tabele stranica i omogućuje nastavak aktivnosti prekinutog procesa.
- Kada **nema slobodne** fizičke stranice, ovaj obrađivač **oslobađa** neku od fizičkih stranica.



# Implementacija upravljanja virtuelnom memorijom

---

- Ako ona sadrži izmenjenu kopiju virtuelne stranice, on pokreće **prenos** ove **kopije** u masovnu memoriju, radi ažuriranja odgovarajuće virtuelne stranice.
- Po završetku ovoga prenosa, obrađivač straničnog prekida usmerava ka oslobođenoj fizičkoj stranici prenos kopije potrebne virtuelne stranice.
- Nakon završetka ovog prenosa i ažuriranja polja odgovarajućih elemenata tabele stranica, on omogućava nastavak aktivnosti prekinutog procesa.

# Implementacija upravljanja virtuelnom memorijom

---

- Za uspešno obavljanje posla, obrađivaču straničnog prekida je potrebna evidencija o položaju virtuelnih stranica u masovnoj memoriji.
- On takođe, koristi i evidenciju slobodnih fizičkih stranica.
- Nju koriste i operacija **zauzimanja** i operacija **oslobađanja**, pa rukovanje ovom evidencijom mora obezbediti sinhronizaciju procesa, i to **onemogućenjem** prekida.
- Operacija **zauzimanja** omogućuje zauzimanje bar **minimalnog** skupa, radi **stvaranja** procesa, a operacija **oslobađanja** oslobađa sve fizičke stranice iz radnog skupa **unišavanog** procesa.

# Osnova sloja za rukovanje virtuelnom memorijom

---

- Sloj za rukovanje virtuelnom memorijom se oslanja na operacije sloja za **rukovanje kontrolerima**, da bi obezbedio prenos kopija virtuelnih blokova na relaciji masovna i radna memorija i da bi smestio adrese svojih obrađivača prekida u odgovarajuće elemente tabele prekida.