

# Upravljanje memorijom



# Pojmovi

- Okvir
  - Blok glavne memorije fiksne duljine.
- Stranica
  - Blok podataka nepromjenjive duljine koji se nalazi u sekundarnoj memoriji (kao što je disk). Stranica podataka može se kopirati u okvir glavne memorije.
- Segment
  - Blok podataka promjenjive duljine koji se nalazi u sekundarnoj memoriji. Cijeli segment može se privremeno kopirati u dostupno područje glavne memorije (segmentacija) ili se segment može podijeliti na stranice, koje se mogu pojedinačno kopirati u glavnu memoriju.
  -

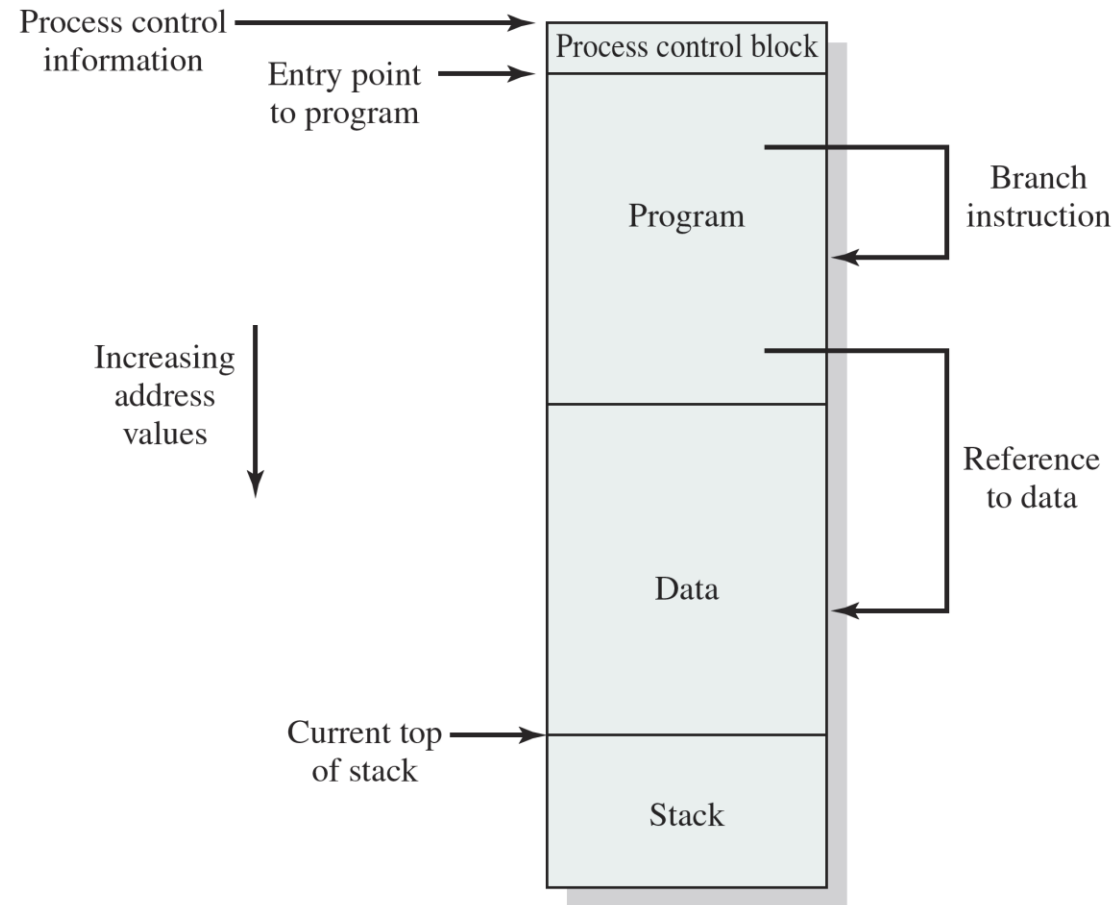
# Preduvjeti za upravljanje memorijom

- Upravljanje memorijom zadovoljava sljedećih zahtjeva:
  - Preseljenje
  - Zaštita
  - Dijeljenje
  - Logička organizacija
  - Fizička organizacija

# Preseljenje

- Programeri obično ne znaju unaprijed koji će drugi programi biti u glavnoj memoriji u vrijeme izvršenja njihova programa
- Aktivni procesi moraju se moći zamijeniti u glavnoj memoriji i izvan nje kako bi se povećala iskorištenost procesora
- Ograničavanje bi bilo da se proces mora smjestiti u isto memorijsko područje kada se ponovno zamijeni
- Možda će trebati premjestiti proces u drugo područje memorije

# Zahtjevi za proces



# Zaštita

- Procesi moraju dobiti dozvolu za upućivanje na memorijska mjesta u svrhu čitanja ili pisanja
- Lokacija programa u glavnoj memoriji je nepredvidiva
- Memorijske reference generirane procesom moraju se provjeriti u vrijeme izvođenja
- Mehanizmi kojima se podupire premještanje također podupiru zaštitu
-

# Dijeljenje

- Korisno je dopustiti svakom procesu pristup istoj kopiji programa, a ne imati vlastitu zasebnu kopiju
- Upravljanje memorijom mora omogućiti kontrolirani pristup zajedničkim područjima memorije bez ugrožavanja zaštite
- Mehanizmi koji se koriste za potporu premještanja podržavaju mogućnosti dijeljenja

# Logička organizacija

- Memorija je organizirana kao linearna

## Programi su napisani u modulima

- Moduli se mogu pisati i sastavljati samostalno
  - Različiti stupnjevi zaštite koji se daju modulima (samo za čitanje, samo za izvršavanje)
  - Zajedničko korištenje na razini modula odgovara korisnikovom načinu gledanja problema
- Segmentacija je alat koji najlakše zadovoljava zahtjeve



# Fizička organizacija

Programeru nije moguće  
ostaviti odgovornost za  
upravljanje memorijom

Memorija dostupna za  
program i njegovi podaci  
možda nisu dovoljni

Programer ne zna koliko će  
prostora biti dostupno

*Preklapanje* omogućuje  
različitim modulima da se  
dodijele istom području  
memorije, ali je dugotrajno  
za programiranje

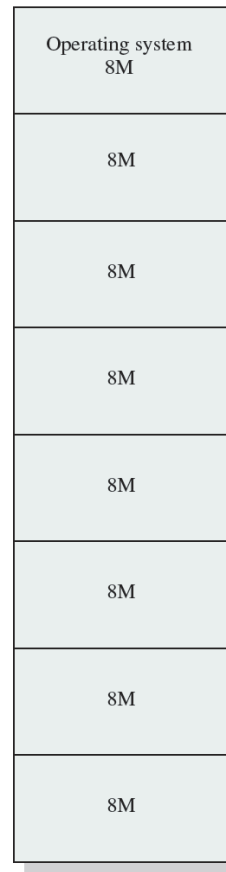
# Particioniranje memorije

- Upravljanje memorijom donosi procese u glavnu memoriju za izvršavanje od strane procesora
  - Uključuje virtualnu memoriju
  - Na temelju segmentacije i stranijenja
- Particioniranje
  - Koristi se u nekoliko varijacija u nekim zastarjelim operativnim sustavima
  - Ne uključuje virtualnu memoriju

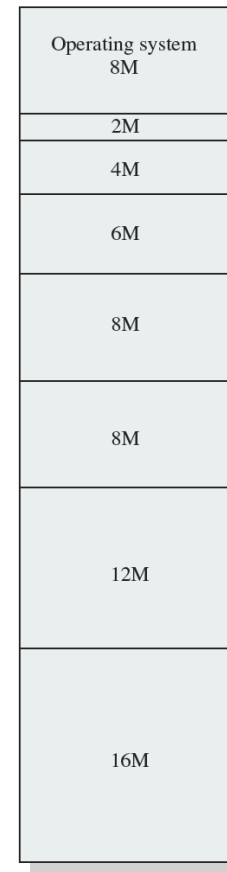
# Tehnike upravljanja memorijom

- Fiksno particioniranje
  - Glavna memorija podijeljena je na brojne statičke particije u vrijeme stvaranja sustava.
- Dinamičko particioniranje
  - Particije se stvaraju dinamički, tako da se svaki proces učitava u particiju potpuno iste veličine kao i taj proces.
- Jednostavna stranična stranica
  - Glavna memorija podijeljena je u nekoliko okvira jednake veličine.
- Jednostavna segmentacija
  - Svaki proces je podijeljen u niz segmenata. Proces preuzima podatke učitavanjem svih njegovih segmenata u dinamičke particije koje ne moraju biti susjedne.
- Stranično povezivanje s virtualnom memorijom
  - Kao i kod jednostavne stranične stranice, osim što nije potrebno učitavati sve stranice procesa. Nerezidentne stranice koje su potrebne automatski se unose kasnije.
- Segmentacija virtualne memorije
  - Kao i kod jednostavne segmentacije, osim što nije potrebno učitavati sve segmente procesa. Nerezidentni segmenti koji su potrebni automatski se unose kasnije.

# Primjer fiksnog particioniranja



(a) Equal-size partitions



(b) Unequal-size partitions

# Nedostaci fiksnih particija jednake veličine

- Program je možda prevelik da stane u particiju
  - Program treba koristiti slojeve
- Korištenje glavne memorije je neučinkovito
  - Bilo koji program, bez obzira na veličinu, zauzima cijelu particiju
  - Unutarnja fragmentacija
    - Izgubljeni prostor zbog toga što je blok učitanih podataka manji od particije

# Nedostaci particija nejednake veličine

- Broj particija u vremenu proizvodnje sustava ograničava broj aktivnih procesa u sustavu
- Procesi s malim zahtjevima neće učinkovito koristiti prostor particije

# Dinamičko particioniranje

- Particije su promjenjive duljine i broja
- Proces se dodjeljuje točno onoliko memorije koliko je potrebno
- Ovu tehniku koristio je IBM-ov operativni sustav mainframe, OS/MVT
-

# Dinamičko particioniranje

## Vanjska fragmentacija

- Memorija postaje sve fragmentiranija
- Potrošnja memorije opada

## Zbijanje

- Tehnika prevladavanja vanjske fragmentacije
- OS mijenja procese tako da su susjedni
- Slobodna memorija je zajedno u jednom bloku
- Dugotrajno s gubitcima procesorskog vremena



# Algoritmi smještaja

## Best-fit

- Odabir bloka koji je po veličini najbliži zahtjevu

## First-fit

- Počinje skenirati memoriju od početka i odabire prvi dostupan blok koji je dovoljno velik

## Next-fit

- Počinje skenirati memoriju s mjesta posljednjeg položaja i odabire sljedeći dostupni blok koji je dovoljno velik

# Buddy sustav

- Sastoji se od fiksnih i dinamičkih shema particioniranja
- Prostor dostupan za dodjelu tretira se kao jedan blok
- Memorijski blokovi dostupni su od riječi veličine  $2K$ ,  
 $L \leq K \leq U$ , gdje je
  - $2^L$  = najmanji blok veličine koji je dodijeljen
  - $2^U$  = najveći blok veličine koji je dodijeljen; općenito  $2^U$  je veličina cijele memorije dostupne za dodjelu

# Preseljenje

- Kada se koristi fiksna shema particija, možemo očekivati da će proces uvijek biti dodijeljen istoj particiji
  - Koja god particija bude odabrana prilikom učitavanja novog procesa, uvijek će se koristiti za zamjenu tog procesa natrag u memoriju nakon što je zamijenjena
  - U tom slučaju može se koristiti jednostavan algoritam za premještanje
  - Kada se proces prvi put učitava, sve relativne memorijske reference u kodu zamjenjuju se apsolutnim glavnim memorijskim adresama određenim osnovnom adresom učitano procesa
- U slučaju particija jednake veličine i u slučaju jednog reda čekanja procesa za particije nejednake veličine, proces može zauzeti različite particije tijekom svog vijeka trajanja
  - Kada se procesna slika prvi put stvori, ona se učitava u neku particiju u glavnoj memoriji; Kasnije se postupak može zamijeniti
  - Kada se naknadno ponovno zamijeni, može se dodijeliti drugoj particiji
  - Isto vrijedi i za dinamičko particioniranje
- Kada se koristi zbijanje, procesi se pomiču dok su u glavnoj memoriji
  - Dakle, lokacije na kojoj su procesi nisu fiksne
  - Promijenit će se svaki put kada se proces zamijeni ili pomakne

# Adrese

## Logičke

- Upućivanje na mjesto memorije neovisno o trenutnoj dodjeli podataka memoriji

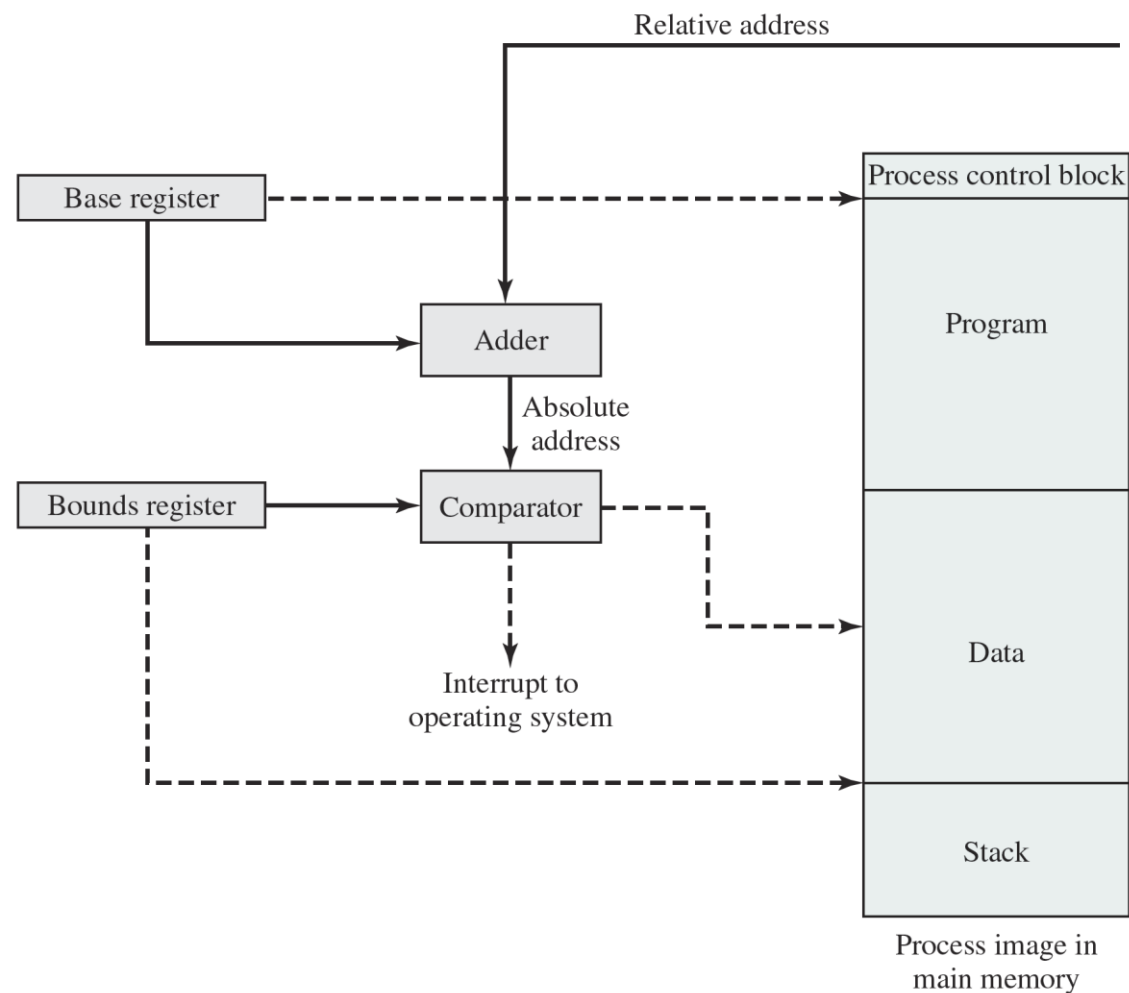
## Relativne

- Poseban primjer logičke adrese, u kojoj je adresa izražena kao mjesto u odnosu na neku poznatu točku

## Fizičke ili apsolutne

- Stvarno mjesto u glavnoj memoriji

# Hardverska podrška za premještanje



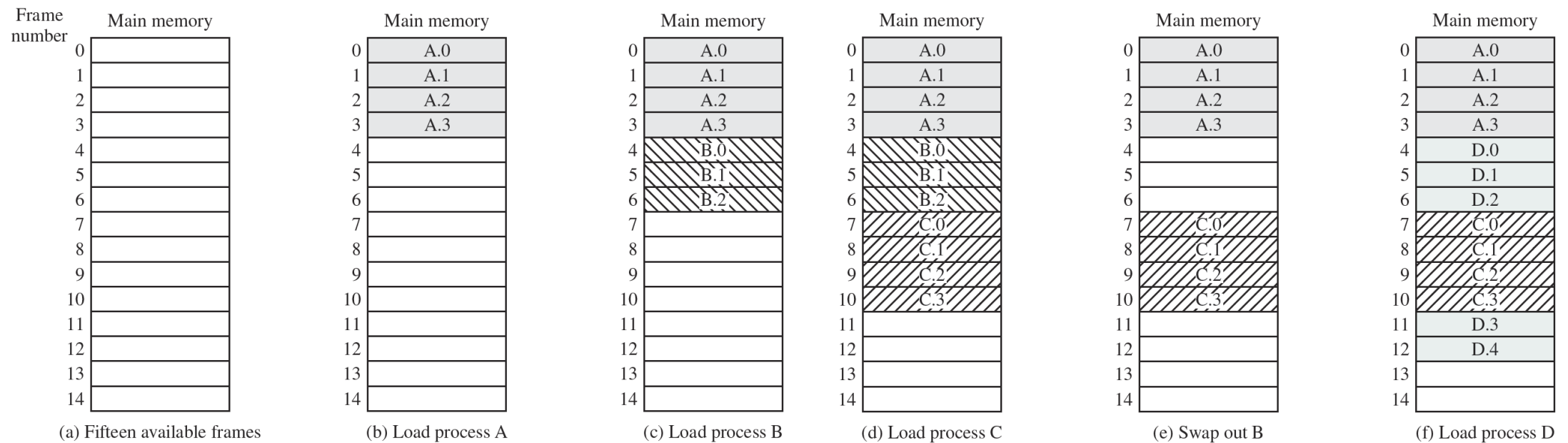
# Straničenje

- Partitioniranje memorije u jednake dijelove fiksne veličine koji su relativno mali
- Proces je također podijeljen na male komade fiksne veličine iste veličine

- 

Stranice	Okviri
<ul style="list-style-type: none"><li>• Dijelovi procesa</li></ul>	<ul style="list-style-type: none"><li>• Dostupni dijelovi memorije</li></ul>

# Dodjela procesa slobodnim okviriima



# Tablica stranica

- Održava je operativni sustav za svaki proces
- Sadrži mjesto okvira za svaku stranicu u procesu
- Procesor mora znati kako pristupiti adresama trenutnog procesa
- Procesor je koristi za izračun fizičke adrese



# Straničenje

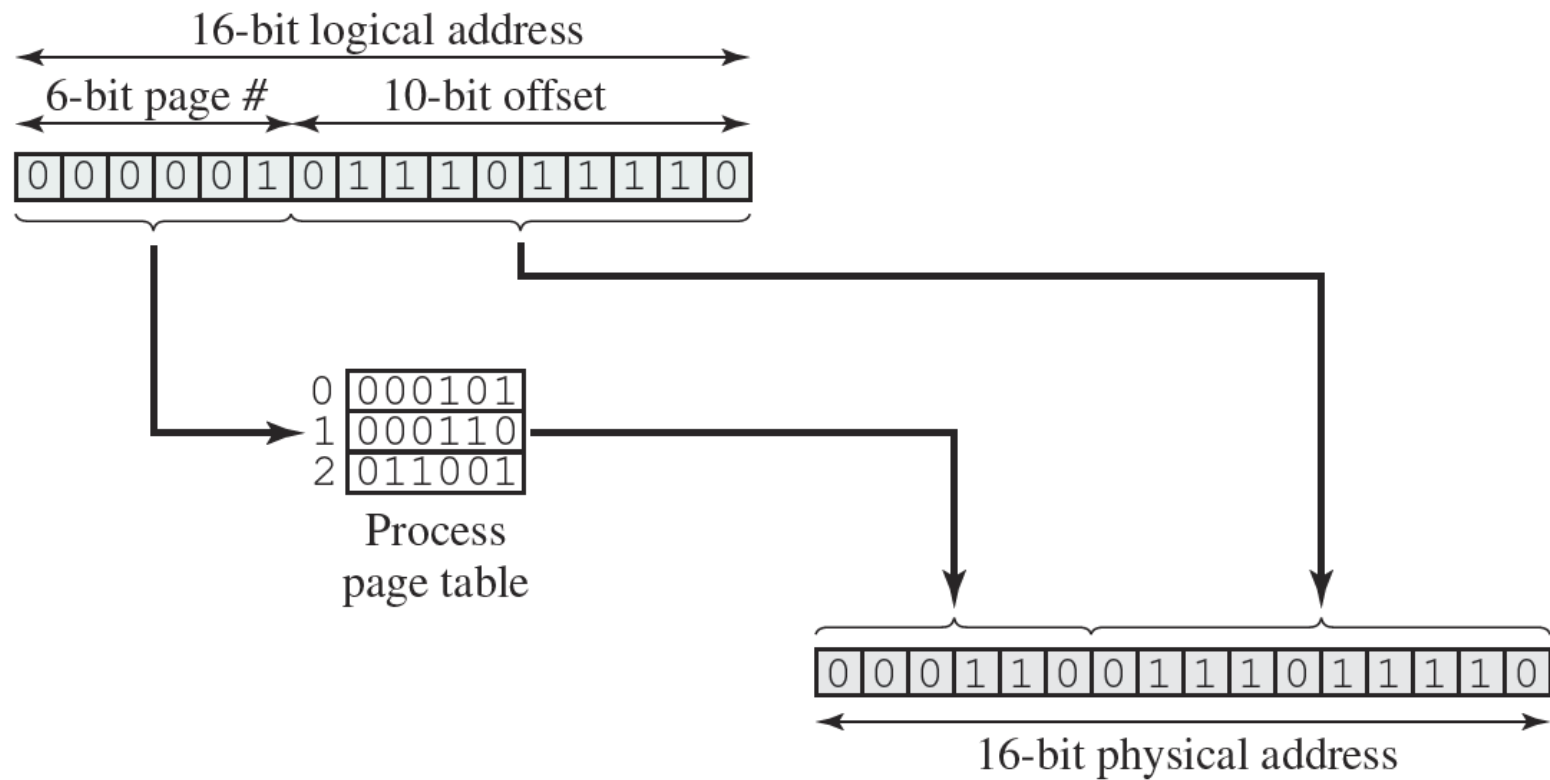


Figure 7.12 Examples of Logical-to-Physical Address Translation

# Segmentacija

- Program se može podijeliti na segmente
  - Može varirati po duljini
  - Postoji maksimalna duljina
- Adresiranje se sastoji od dva dijela:
  - Broj segmenta
  - Pomak
- Slično dinamičkom particioniranju
- Uklanja unutarnju fragmentaciju
-

# Segmentacija

- Obično vidljiva
- Pruža se kao pogodnost za organizaciju programa i podataka
- Programer će obično dodijeliti programe i podatke različitim segmentima
- U svrhu modularnog programiranja program ili podaci mogu se dalje raščlaniti na više segmenata
  - Glavna nedostatak ovog pristupa je da programer mora biti svjestan maksimalnog ograničenja veličine segmenta

# Prijevođenje adresa

- Još jedna posljedica nejednakih veličina segmenata je da ne postoji jednostavan odnos između logičkih adresa i fizičkih adresa
- Za prijevod adrese potrebni su sljedeći koraci:

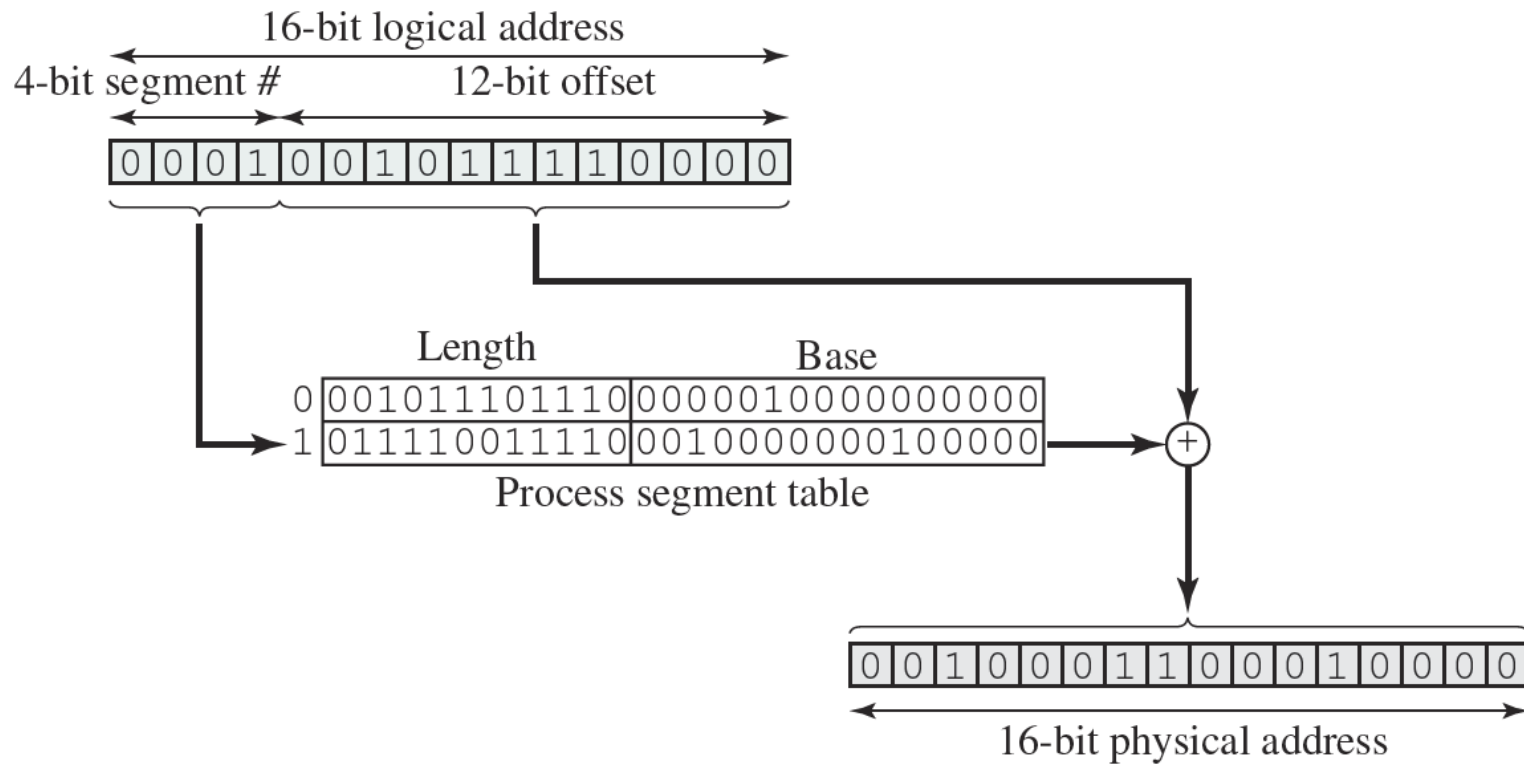
Izdvoji broj segmenta kao krajnje lijeve  $n$  bitove logičke adrese

Pomoću broja segmenta kao indeksa u tablici segmenta procesa pronađite početnu fizičku adresu segmenta

Usporedite pomak, izražen u najdesnijim  $m$  bitovima, s duljinom segmenta. Ako je pomak veći ili jednak duljini, adresa nije valjana

Željena fizička adresa je zbroj početne fizičke adrese segmenta plus pomak

# Segmentacija



# Virtualne memorije

- Virtualna memorija
  - Shema dodjele prostora za pohranu u kojoj se sekundarna memorija može riješiti kao da je dio glavne memorije.
- Virtualna adresa
  - Adresa dodijeljena lokaciji u virtualnoj memoriji kako bi se omogućilo pristup tom mjestu kao da je dio glavne memorije.
- Virtualni adresni prostor
  - Virtualna pohrana dodijeljena procesu.
- Adresni prostor
  - Raspon memorijskih adresa dostupnih procesu.
- Stvarna adresa
  - Adresa mjesta za pohranu u glavnoj memoriji.

# Hardverske i upravljačke strukture

- Dvije karakteristike temeljne za upravljanje memorijom:
  - Sve memorijske reference su logičke adrese koje se dinamički prevode na fizičke adrese u vrijeme izvođenja
  - Proces se može razbiti na brojne dijelove koji ne moraju biti smješteni jedan pored drugog u glavnoj memoriji tijekom izvršavanja
- Ako su ove dvije karakteristike prisutne, nije potrebno da sve stranice ili segmenti procesa budu u glavnoj memoriji tijekom izvršavanja

# Izvršavanje procesa

- Operativni sustav u glavnu memoriju unosi nekoliko dijelova programa
- Rezidentni skup
  - Dio procesa koji je u glavnoj memoriji
- Prekid se generira kada je potrebna adresa koja nije u glavnoj memoriji
- Operacijski sustav smješta proces u stanje blokiranja



# Izvršavanje procesa

- Dio procesa koji sadrži logičku adresu unosi se u glavnu memoriju
  - Operacijski sustav izdaje zahtjev za ulazno/izlazno čitanje diska
  - Drugi proces se šalje za pokretanje tijekom uključivanja/izlaza na disku
  - Prekid se izdaje kada je ulazno/izlazna operacija dovršena, zbog čega operativni sustav postavlja zahvaćeni proces u stanje Spreman (Ready)

# Implikacije

- Više procesa može se održavati u glavnoj memoriji
  - Budući da su učitani samo neki dijelovi bilo kojeg određenog procesa, ima mjesta za više procesa
  - To dovodi do učinkovitijeg korištenja procesora jer je vjerojatnije da će barem jedan od brojnih procesa biti u spremnom stanju u bilo kojem određenom trenutku
- Proces može biti veći od svih glavne memorije
  - Ako je program koji se piše prevelik, programer mora osmisliti načine za strukturiranje programa u dijelove koji se mogu učitati odvojeno u nekoj vrsti strategije prekrivanja
  - S virtualnom memorijom koja se temelji na straničanju ili segmentaciji, taj je posao prepušten OS-u i hardveru
  - OS automatski učitava dijelove procesa u glavnu memoriju prema potrebi

# Stvarna i virtualna memorija



# Pogađanje (*Thrashing*)



# Načelo lokaliteta

- Reference programa i podataka unutar procesa obično se grupiraju
- Samo nekoliko dijelova procesa bit će potrebno u kratkom vremenskom razdoblju
- Stoga je moguće inteligentno nagađati koji će komadi biti potrebni u budućnosti
- Izbjegava pogađanje

# Podrška potrebna za virtualnu memoriju

Da bi virtualna memorija bila praktična i učinkovita:

- Hardver mora podržavati straničenje i segmentaciju
- Operativni sustav mora uključivati softver za upravljanje prebacivanjem stranica i/ili segmenata između sekundarne memorije i glavne memorije

# Straničenje

- Pojam virtualna memorija obično je povezan sa sustavima koji koriste straničenje
- Korištenje straničenja za postizanje virtualne memorije prvi put je realizirano za računalo Atlas
- Svaki proces ima svoju tablicu stranica
  - Svaki unos tablice stranice (*PTE - page table entry*) sadrži broj okvira odgovarajuće stranice u glavnoj memorij
  - Tablica stranice također je potrebna za shemu virtualne memorije koja se temelji na straničanju

# Tipični oblici upravljanja memorijom

Virtual address



Page table entry



(a) Paging only

Virtual address

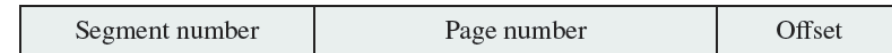


Segment table entry



(b) Segmentation only

Virtual address



Segment table entry



Page table entry

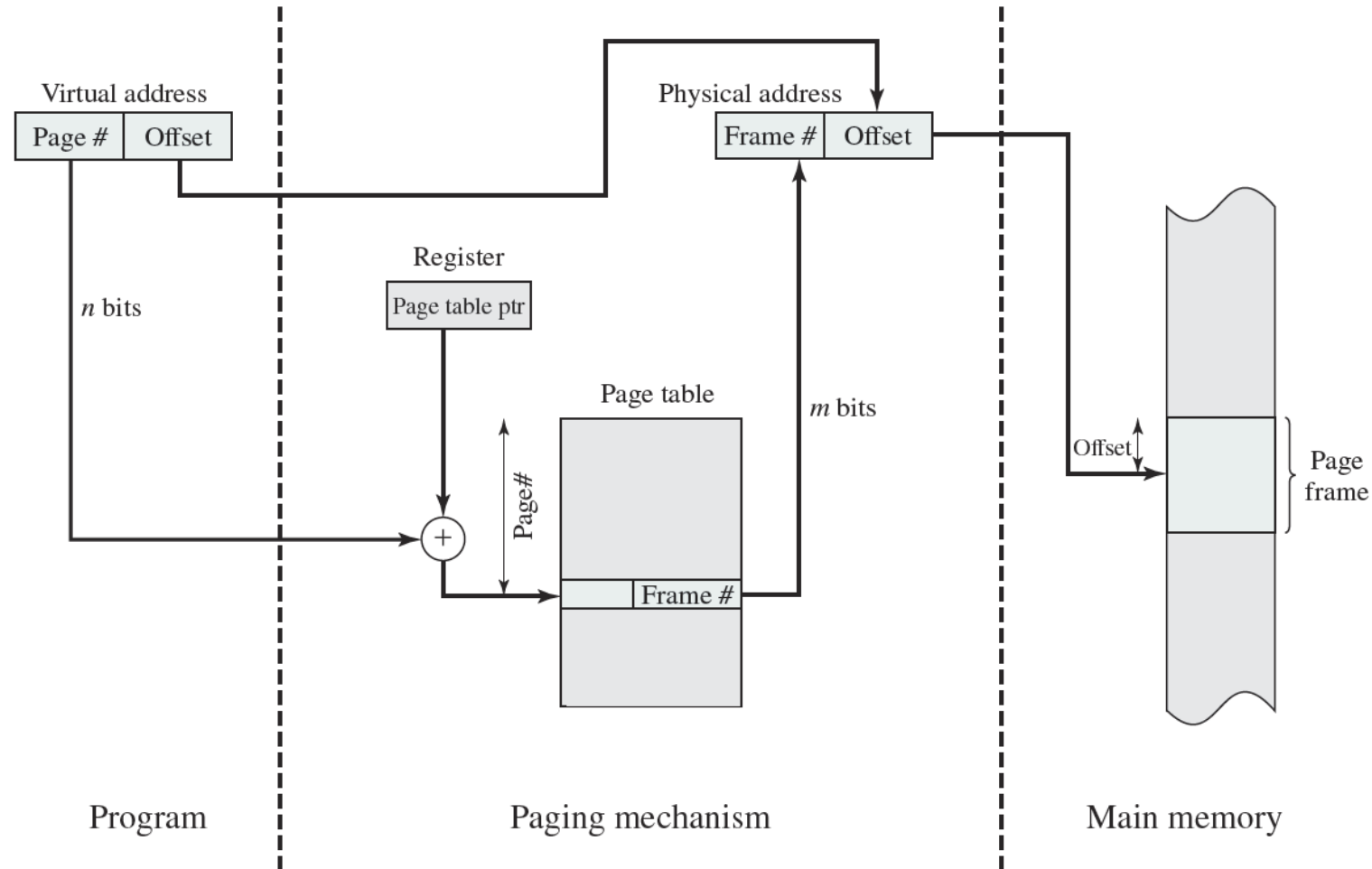


(c) Combined segmentation and paging

P = present bit  
M = modified bit



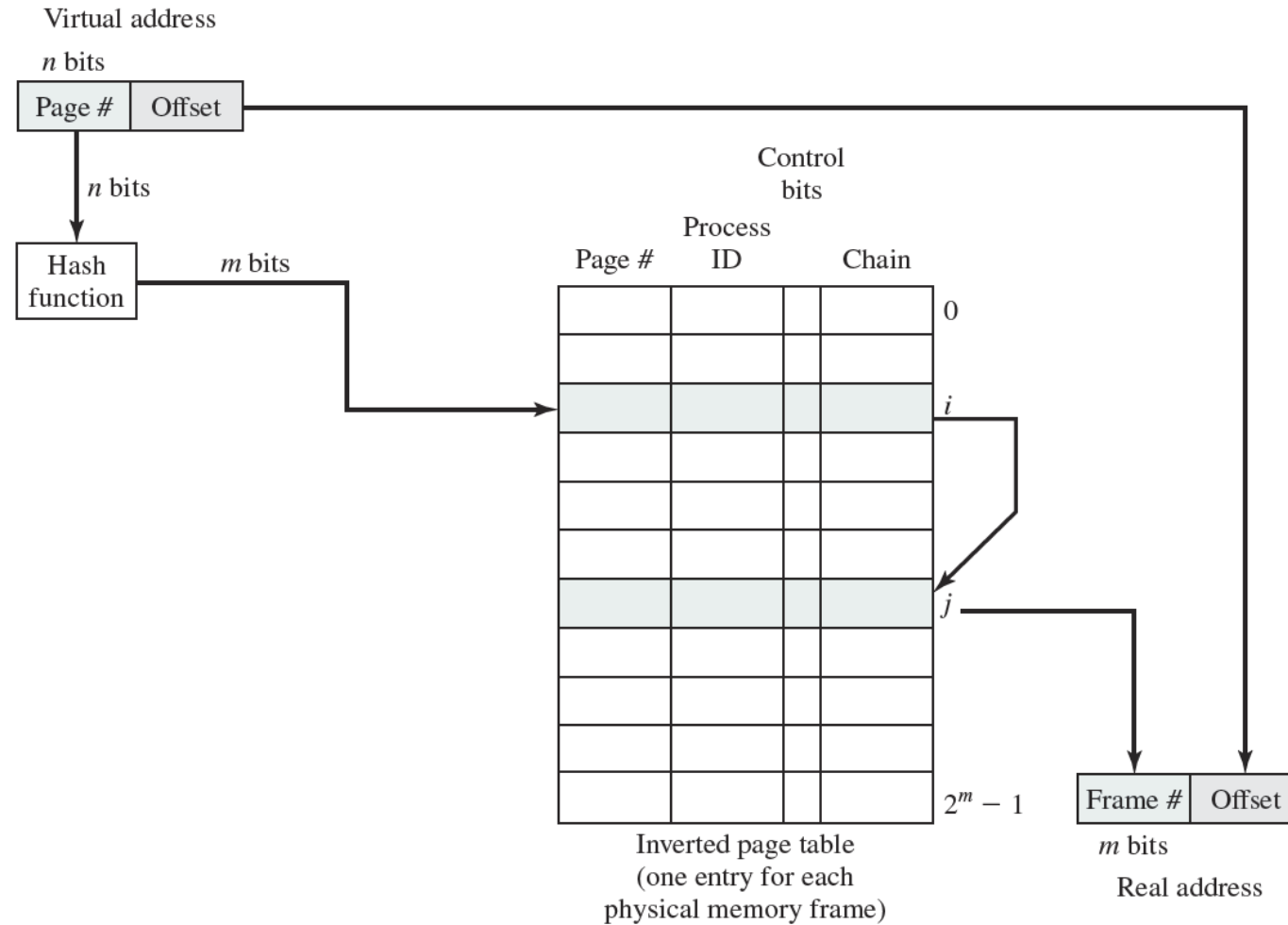
# Prijevod adresa u straničnom sustavu



# Inverzna tablica stranica

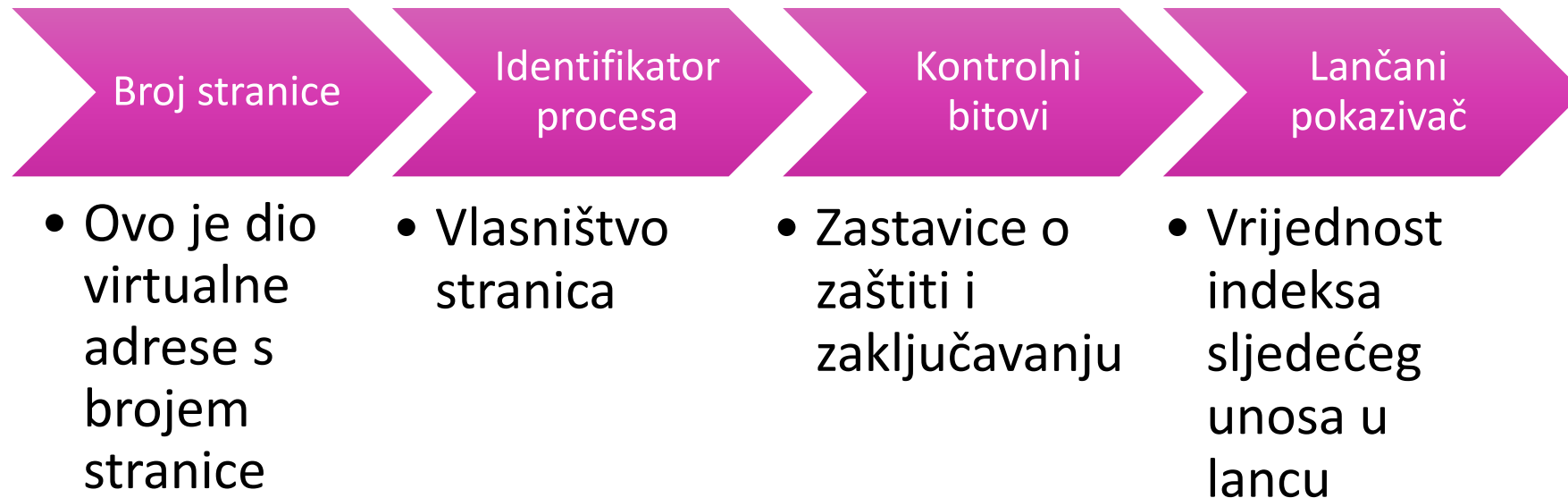
- Dio virtualne adrese s brojem stranice mapiran je u HASH vrijednost
  - HASH vrijednost upućuje na obrnutu tablicu stranice
- Fiksni udio stvarne memorije potreban je za tablice bez obzira na broj podržanih procesa ili virtualnih stranica
- Struktura se naziva inverzna jer indeksira unose tablice stranica prema broju okvira, a ne prema virtualnom broju stranice

# Struktura inverzne tablice stranice



# Inverzna tablica stranica

- Svaki unos u tablici stranice uključuje:

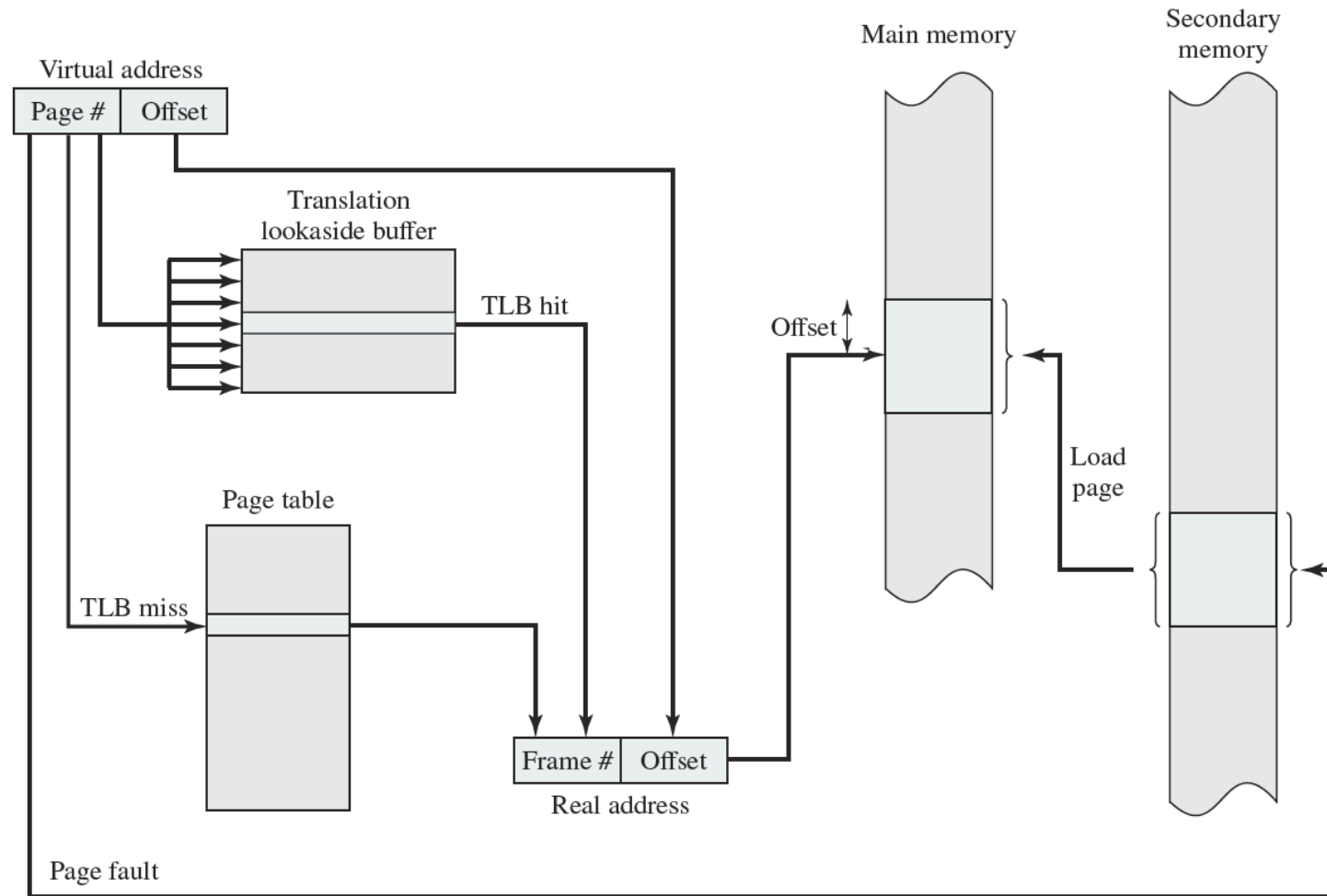


# Međuspremnik prevođenja (TLB)

## Translation Lookaside Buffer (TLB)

- Svaka referenca virtualne memorije može uzrokovati dva pristupa fizičkoj memoriji:
  - Jedan za dohvaćanje unosa tablice stranice
  - Jedan za dohvaćanje podataka
- Kako bi se prevladao učinak udvostručenja vremena pristupa memoriji, većina virtualnih memorijskih shema koristi posebnu predmemoriju velike brzine koja se naziva međuspremnik prijevođenja (TLB)
  - Ova predmemorija funkcionira na isti način kao i predmemorija memorije i sadrži one entitete tablice stranice koji su nedavno korišteni

# Korištenje međuspremnik prevodenja



# Asocijativno mapiranje

- TLB sadrži samo neke unose u tablici stranica, tako da ne možemo jednostavno indeksirati u TLB na temelju broja stranice
  - Svaki TLB unos mora sadržavati broj stranice, kao i cijeli unos tablice stranice
- Procesor je opremljen hardverom koji mu omogućuje istovremeno ispitivanje brojnih TLB unosa kako bi se utvrdilo postoji li podudaranje za broj stranice

# Veličina stranice

- Što je manja veličina stranice, manja je količina unutarnje fragmentacije
  - Međutim, potrebno je više stranica po procesu
  - Više stranica po procesu znači veće tablice stranica
  - Za velike programe u jako višeprogramiranom okruženju neki dio tablica stranica aktivnih procesa mora biti u virtualnoj memoriji umjesto u glavnoj memoriji
  - Fizičke karakteristike većine sekundarno-memorijskih uređaja pogoduju većoj veličini stranice za učinkovitiji blok prijenos podataka



# Veličina stranice

- Suvremene tehnike programiranja koje se koriste u velikim programima imaju tendenciju smanjenja lokalnih referenci unutar procesa

Pitanje dizajna veličine stranice odnosi se na veličinu fizičke glavne memorije i veličinu programa



Glavna memorija postaje sve veća, a adresni prostor koji koriste aplikacije također raste



Najočitije na osobnim računalima gdje aplikacije postaju sve složenije

# Segmentacija

- Segmentacija omogućuje programeru da vidi memoriju koja se sastoji od više adresnih razmaka ili segmenata
- Prednosti:
  - Pojednostavljuje rukovanje rastućim strukturama podataka
  - Omogućuje samostalnu izmjenu i ponovnu kompenziju programa
  - Omogućuje dijeljenje podataka među procesima
  - Omogućuje zaštitu podataka

# Organizacija segmenta

- Svaka stavka tablice segmenta sadrži početnu adresu odgovarajućeg segmenta u glavnoj memoriji i duljinu segmenta
- Potrebno je malo kako bi se utvrdilo je li segment već u glavnoj memoriji
- Potreban je još jedan bit kako bi se utvrdilo je li segment izmijenjen otkad je učitao u glavnu memoriju

# Kombiniranje straničenja i segmentacije

U kombiniranom sustavu straničenja/segmentacije adresni prostor korisnika raspoređen je na više segmenata. Svaki segment podijeljen je na brojne stranice fiksne veličine koje su po duljini jednake glavnom memorijskom okviru

Segmentacija je vidljiva programeru

Straničenje je transparentno za programera

# Zaštita i zajedničko korištenje

- Segmentacija se podvrgava provedbi politika zaštite i dijeljenja
- Svaki unos ima osnovnu adresu i duljinu tako da se nenamjernim pristupom memoriji može kontrolirati
- Dijeljenje se može postići segmentima koji se odnose na više procesa

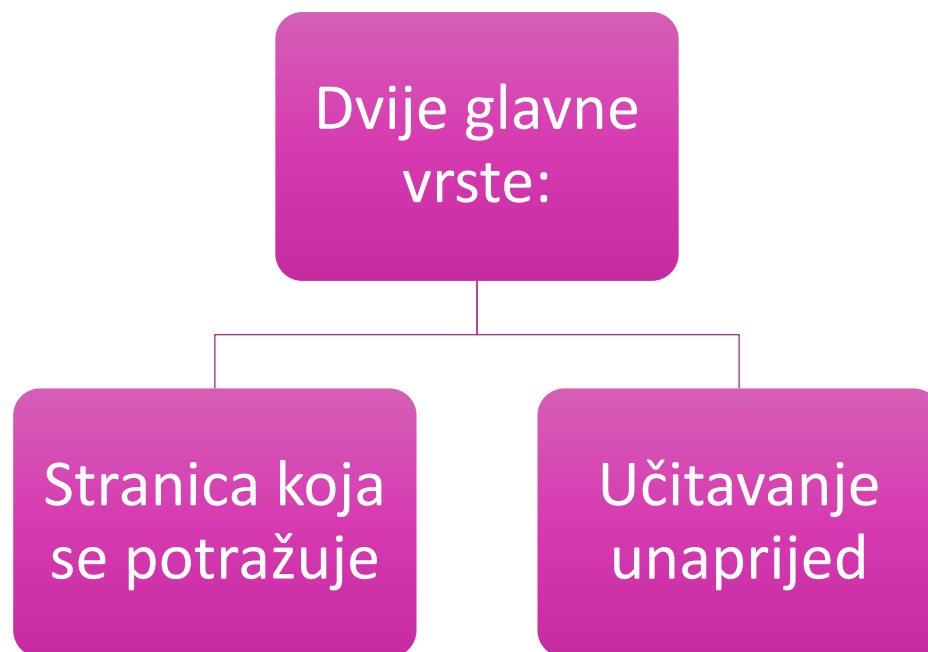
# Softver operacijskog sustava

Dizajn dijela operativnog sustava za upravljanje memorijom ovisi o tri temeljna područja izbora:

- Treba li koristiti tehnike virtualne memorije ili ne
- Korištenje straničenja ili segmentacije ili oboje
- Algoritmi korišteni za različite aspekte upravljanja memorijom

# Pravila dohvaćanja

- Određuje kada stranicu treba unijeti u memoriju



# Stranica koja se potražuje

- Unosi stranice u glavnu memoriju samo kada se upućuje na mjesto na stranici
- Greške na stranici prilikom prvog pokretanja procesa
- Načelo lokaliteta sugerira da će, kako se donosi sve više stranica, većina budućih referenci biti na stranice koje su nedavno učitane, a greške na stranicama trebale bi pasti na vrlo nisku razinu



# Učitavanje unaprijed

- Unose se stranice koje nisu one koje zahtijeva greška stranice
- Iskorištava karakteristike većine sekundarnih memorijskih uređaja
- Ako su stranice procesa pohranjene istodobno u sekundarnoj memoriji, učinkovitije je učitati brojne stranice odjednom
- Neučinkovito ako se ne budu koristile dodatne stranice
- Ne treba miješati s "zamjenom"

# Pravila smještaja

- Određuje gdje će se u stvarnoj memoriji nalaziti dio procesa
- Važno pitanje dizajna u segmentacijskom sustavu
- Straničenje ili kombinirano straničenje sa segmentacijom nije važno jer hardver obavlja funkcije s jednakom učinkovitošću
- Za NUMA sustave poželjna je strategija automatskog smještaja

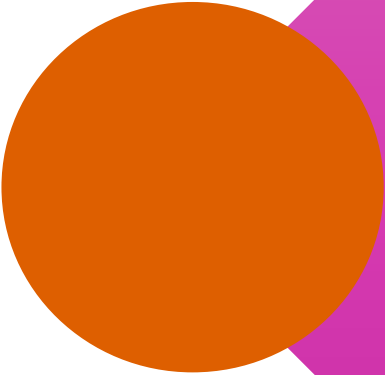
# Pravila zamjene

- Bavi se odabirom stranice u glavnoj memoriji koja će se zamijeniti kada se mora učitati nova stranica
  - Cilj je da stranica koja je uklonjena bude stranica na koju je najmanje vjerojatno da će se referencirati u bliskoj budućnosti
  - Što je politika zamjene složenija, to su veći troškovi hardvera i softvera za njegovu implementaciju

# Zaključavanje okvira

- Kada je okvir zaključan, stranica koja je trenutno pohranjena u tom okviru možda neće biti zamijenjena
  - Jezgra OS-a, kao i ključne upravljačke strukture drže se u zaključanim okvirima
  - Ulazno-izlazni međuspremnik i vremenski kritična područja mogu biti zaključani u memorijske okvire
  - Zaključavanje se postiže postavljanjem brave na okvire

# Osnovni algoritmi



Algoritmi koji se koriste za odabir stranice za zamjenu:

- Optimalni
- First-in-first-out (FIFO)
- Least recently used (LRU)
- Least frequently used (LFU)
- Clock

# Ponašanje algoritama zamjene stranice

Page address stream

2 3 2 1 5 2 4 5 3 2 5 2

OPTIMAL

2	2	2	2	2	2	4	4	4	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
			1	5	5	5	5	5	5	5	5
				F		F			F		

FIFO

2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	3	2	2	2	2	2	5	5
			1	1	1	4	4	4	4	4	2
				F	F	F		F		F	F

LRU

2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	5	5	5	5	5	5	5	5
			1	1	1	4	4	4	2	2	2
				F		F		F	F		

LFU

2	2	2*	2*	2*	2**	2**	2**	2**	2***	2***	2****
	3	3	3	5	5	5	5	5	5	5	5
			1	1	1	4	4	3	3	3	3
				F		F		F			

CLOCK

2*	2*	2*	2*	5*	5*	5*	5*	3*	3*	3*	3*
	3*	3*	3*	3	2*	2*	2*	2	2*	2	2*
			1*	1	1	4*	4*	4	4	5*	5*
				F		F		F		F	

# First-in-First-out (FIFO)

- Okvire stranica dodijeljene procesu tretira kao kružni međuspremnik
- Stranice se uklanjaju u stilu kružnog bojanja
  - Jednostavna politika zamjene za provedbu
- Stranica koja je najduže u memoriji se zamijenjuje

# Least Recently Used (LRU)

- Zamjenjuje stranicu koju se najduže ne koristi
- Po načelu lokaliteta, ovo bi trebala biti stranica za koju je najmanje vjerojatno da će se koristiti u bliskoj budućnosti
- Teško se provodi
  - Jedan od pristupa je označavanje svake stranice s vremenom posljednje reference
    - Zahtijeva vrijeme za obradu

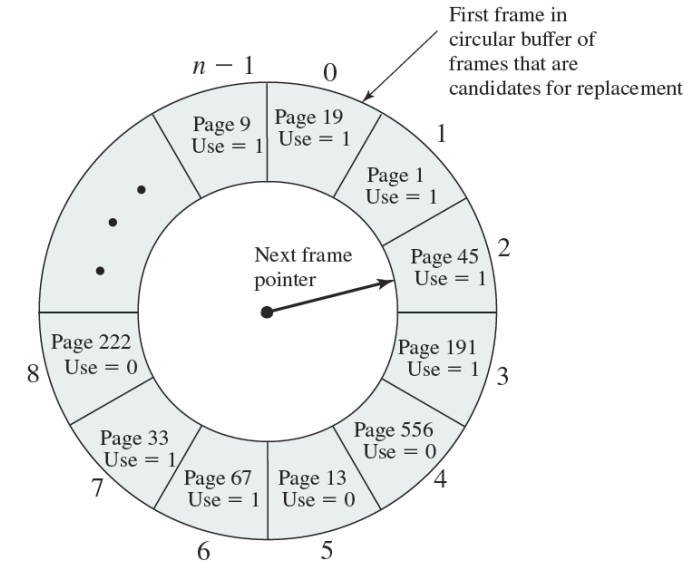


# Least Frequently Used (LFU)

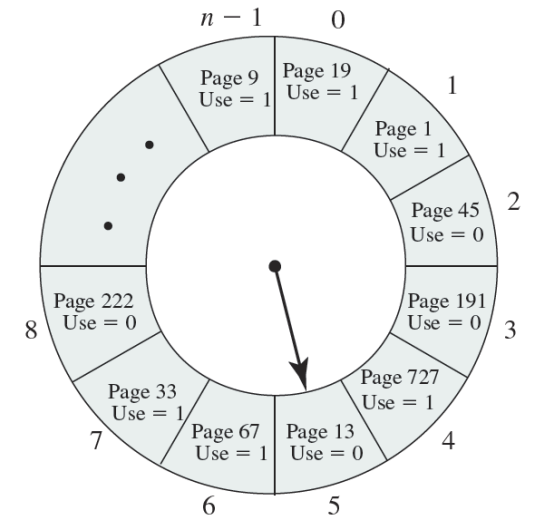
- Zamjenjuje stranicu najnižom referentnom frekvencijom
- Ovo bi trebala biti stranica za koju je najmanje vjerojatno da će se koristiti u bliskoj budućnosti
- U situaciji kada moramo birati između istih frekvencija, trebamo koristiti LRU algoritam
- Također, teško ga je provesti
-

# Clock Policy

- Zahtijeva povezivanje dodatnog bita sa svakim okvirom
  - Naziva se bit upotrabe
- Kada se stranica prvi put učitava u memoriju ili referencira, bit korištenja postavljen je na 1
- Skup okvira smatra se kružnim međuspremnikom
- Algoritam prenosi bilo koji okvir s korisnim bitom 1
- Okviri stranica kao raspoređeni u krugu



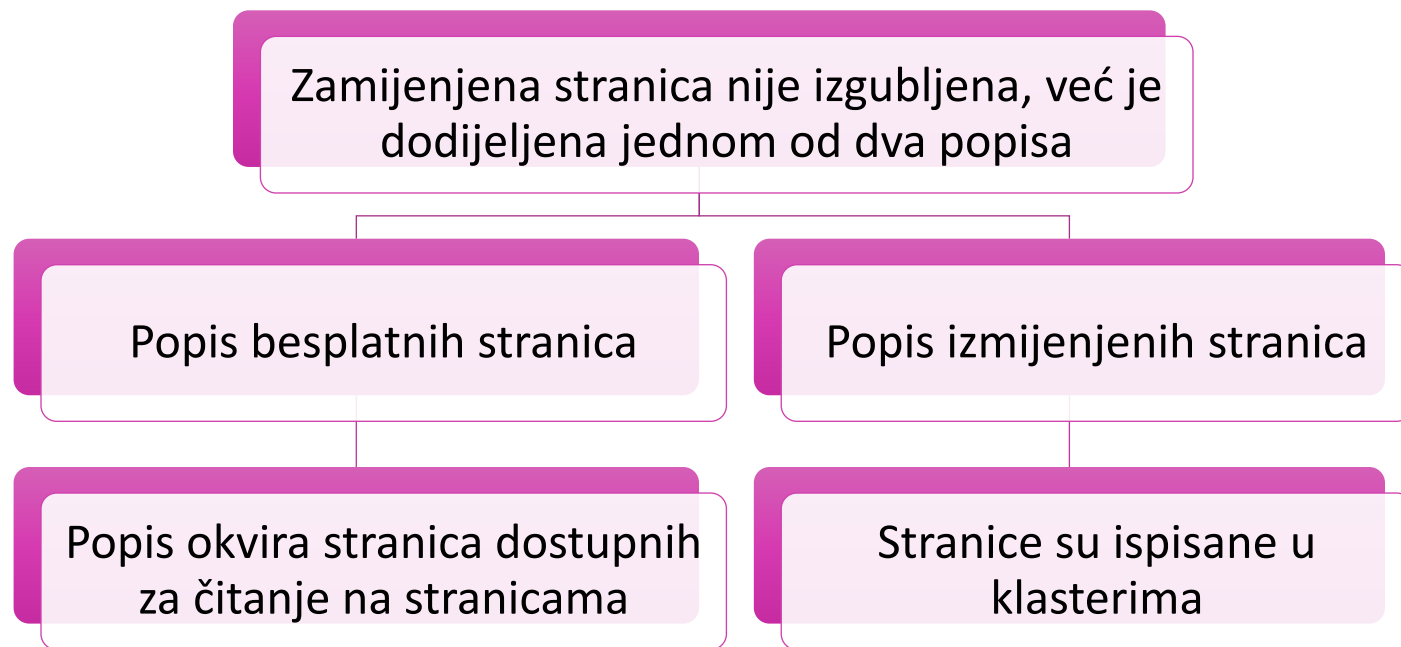
(a) State of buffer just prior to a page replacement



(b) State of buffer just after the next page replacement

# Međuspremnik stranica

- Poboljšava performanse straničenja i omogućuje korištenje jednostavnijeg pravila zamjene stranice



# Pravila zamjene i veličina predmemorije

- Uz velike predmemorije, zamjena stranica može imati učinak na performanse
  - Ako se okvir stranice odabran za zamjenu nalazi u predmemoriji, taj se blok predmemorije gubi kao i stranica koju drži
  - U sustavima koji koriste međuspremnik stranica performanse predmemorije mogu se poboljšati pravilima za smještaj stranice u međuspremniku stranica
  - Većina operacijskih sustava postavlja stranice odabirom proizvoljnog okvira stranice iz međuspremnika stranica

# Kontrola opterećenja

- Određuje broj procesa koji će biti rezidentni u glavnoj memoriji
  - Razina multiprogramiranja
- Kritično u učinkovitom upravljanju memorijom
- Premalo procesa, mnogo prilika kada će svi procesi biti blokirani i puno vremena će se potrošiti na zamjenu
- Previše procesa dovest će do pogađanja

# Obustava procesa

- Ako se želi smanjiti stupanj multiprogramiranja, jedan ili više trenutčno rezidentnih procesa mora se zamijeniti

Postoji šest mogućnosti:

- Proces najnižeg prioriteta
- Procesi s greškom
- Zadnji aktiviran proces
- Postupak s najmanjim rezidentnim skupom
- Najveći proces
- Postupak s najvećim preostalim prozorom izvršenja

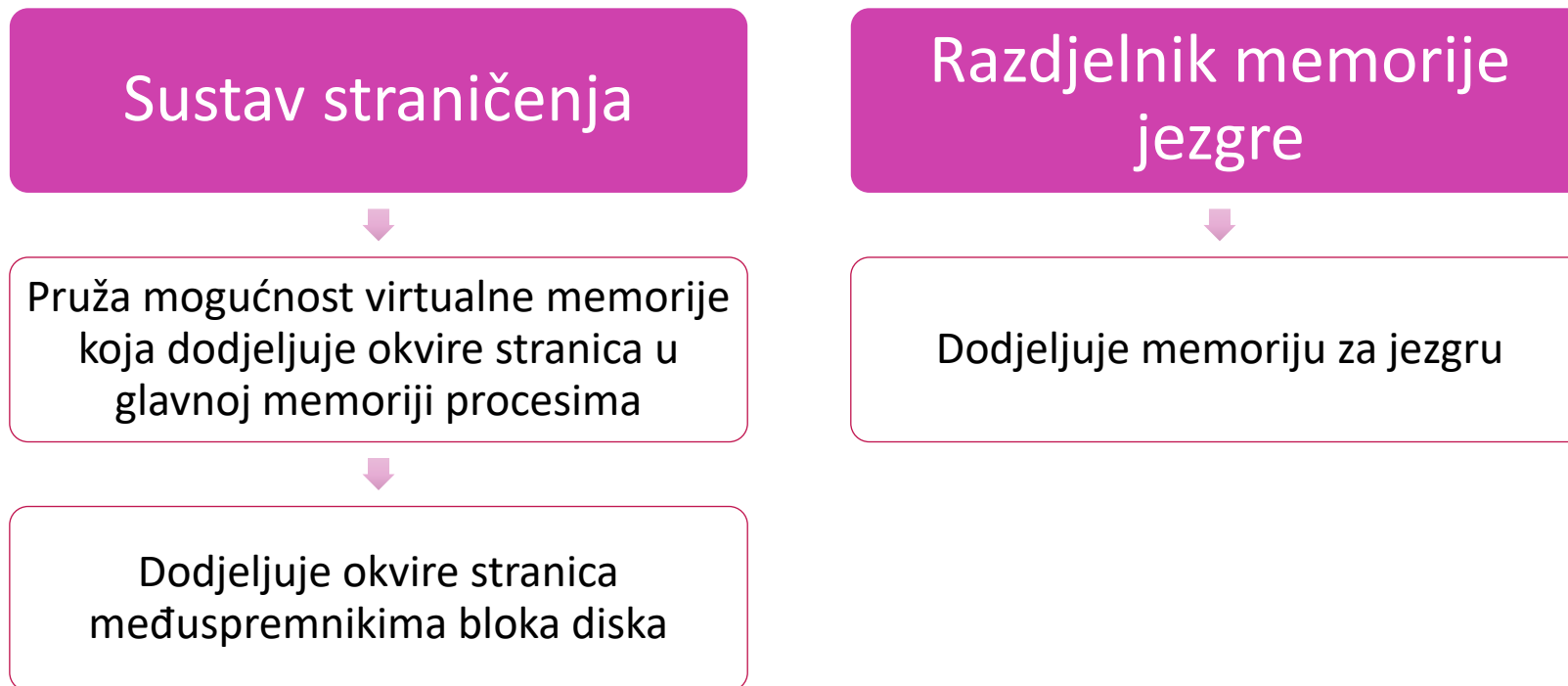
# Upravljanje memorijom - UNIX

- Namijenjeno da bude strojno neovisno, tako da će se njegove sheme upravljanja memorijom razlikovati
  - Rani UNIX: varijabilno particioniranje bez sheme virtualne memorije
  - Trenutne implementacije UNIX-a i Solarisa koriste virtualnu memoriju sa stranicama

SVR4 i Solaris koriste dvije zasebne sheme:

- Sustavstraničenja
- Aloktor memorije jezgre

# Sustav straničenja i razdjelnik memorije jezgre





# Zamjena stranica

- Tablica podataka okvira stranice koristi se za zamjenu stranice
- Pokazivači se koriste za stvaranje popisa unutar tablice
  - Svi dostupni okviri povezani su na popisu praznih okvira dostupnih za dovođenje stranica
  - Kada broj dostupnih okvira padne ispod određenog praga, jezgra će zauzeti brojne okvire kako bi to nadoknadila

# Upravljanje memorijom - Linux

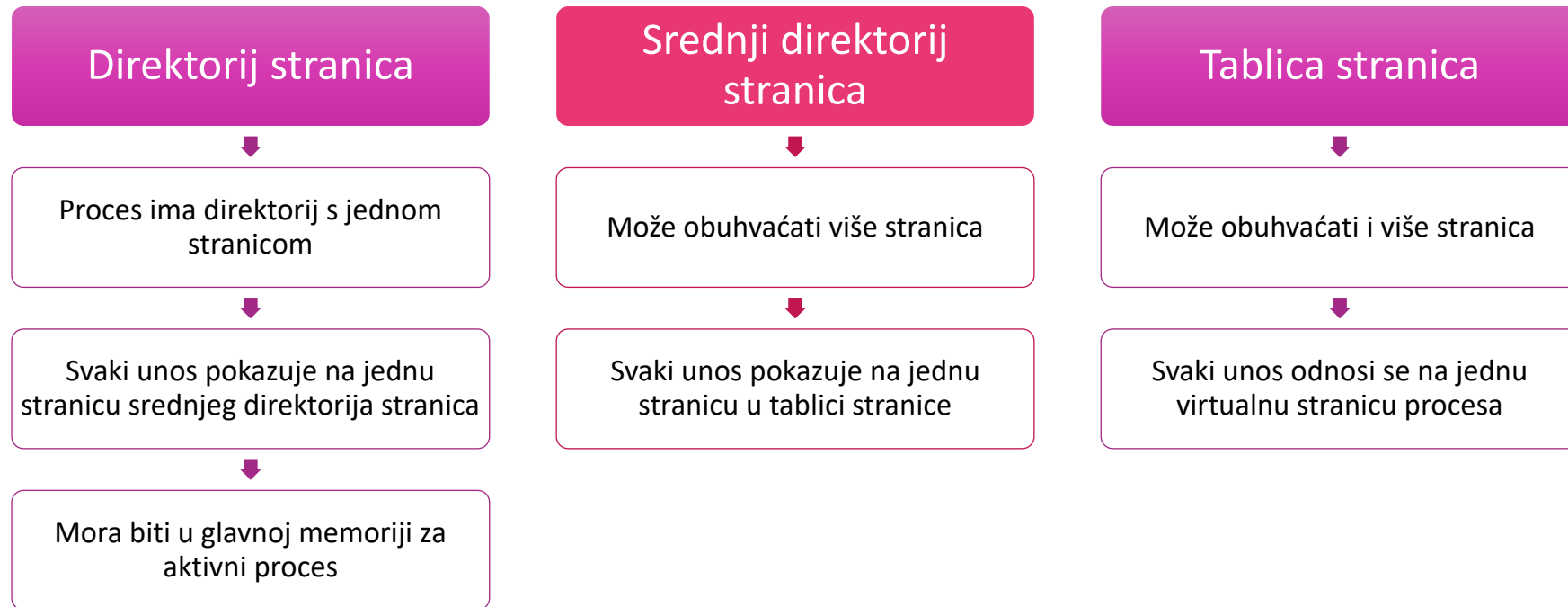
- Dijeli mnoge karakteristike s UNIX-om
- Prilično je složeno

Dva glavna  
aspekta

- Procesna virtualna memorija
- Dodjela memorije jezgre

# Linux virtualna memorija

- Struktura tablice stranice s tri razine:



# Linux - zamjena stranica

- Na temelju algoritma sata
- Bit upotrebe zamjenjuje se 8-bitnom varijablom starosti
  - Povećava se svaki put kada se pristupi stranici
- Povremeno smanjuje dobne bitove
  - Stranica s dobi od 0 godina je "stara" stranica na koju se već neko vrijeme ne spominje i najbolji je kandidat za zamjenu
- Oblik najčešće korištenih pravila

# Upravljanje memorijom u sustavu Windows

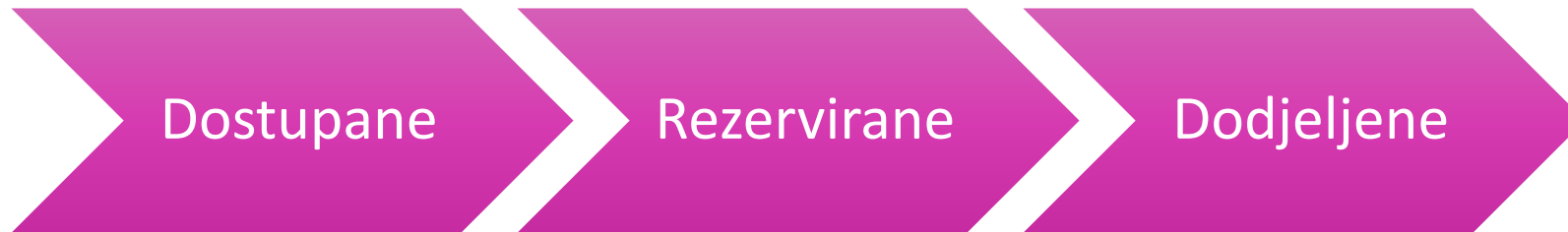
- Upravitelj virtualne memorije kontrolira kako se memorija dodjeljuje i kako se izvodi straničavanje
- Dizajniran za rad na raznim platformama
- Koristi veličine stranica u rasponu od 4 Kbytes do 64 Kbytes

# Mapiranje virtualne adrese - Windows

- Na 32-bitnim platformama svaki korisnički proces vidi zaseban 32-bitni adresni prostor koji omogućuje 4 GB virtualne memorije po procesu
  - Prema zadanim postavkama polovica rezervirana je za OS
  - Velike memorijski intenzivne aplikacije rade učinkovitije pomoću 64-bitnog sustava Windows
  - Većina modernih računala koristi arhitekturu 64bitnih procesora koja se može pokretati kao 32-bitni ili 64-bitni sustav

# Windows straničenje

- Prilikom stvaranja proces može iskoristiti cijeli korisnički prostor od gotovo 2 GB
- Taj je prostor podijeljen na stranice fiksne veličine od 64 KB
- Regije mogu biti u jednoj od tri stanja:



# Rezidentni sustav upravljanja skupovima

- Windows koristi varijabilnu alokaciju, lokalni doseg
- Kada se aktivira, procesu se dodjeljuje struktura podataka za upravljanje radnim skupom
- Radni skupovi aktivnih procesa prilagođavaju se ovisno o dostupnosti glavne memorije



# Androida - upravljanje memorijom

- Android uključuje niz proširenja na uobičajeni Linux kernel objekt za upravljanje memorijom
- To uključuje:
  - ASHMem
    - Ova značajka pruža anonimnu zajedničku memoriju koja apstrahira memoriju kao opisnike datoteka
    - Deskriptor datoteke može se proslijediti drugom procesu za zajedničko korištenje memorije
  - Pmem
    - Ova značajka dodjeljuje virtualnu memoriju tako da je fizički susjedna
    - Ova je značajka korisna za hardver koji ne podržava virtualnu memoriju
  - Low Memory Killer
    - Ova značajka omogućuje sustavu da obavijesti aplikaciju ili aplikacije da im mora uzeti dio memorije
    - Ako aplikacija ne otpusti dio memorije OS je prekida

# Sažetak

- Preduvjeti za upravljanje memorijom
  - Preseljenje
  - Zaštita
  - Dijeljenje
  - Logička organizacija
  - Fizička organizacija
- Straničenje
- Partitioniranje memorije
  - Fiksno partitioniranje
  - Dinamičko partitioniranje
  - Buddy sustav
  - Preseljenje
- Segmentacija
- Hardverske i upravljačke strukture
  - Lokalitet i virtualna memorija
  - Kombinirano straničenje i segmentacija
  - Zaštita i zajedničko korištenje
- Programi OS-a
  - Pravilo dohvaćanja
  - Pravila smještaja
  - Pravila zamjene
  - Upravljanje rezidentnim skupom
- UNIX upravljanje memorijom
  - Stranični sustav
- Linux - upravljanje memorijom
  - Linux virtualna memorija
  - Dodjela memorije jezgre
- Windows - upravljanje memorijom
  - Mapiranje virtualnih adresa sustava Windows
  - Windows straničenje
  - Zamjena u Windows sustavima
- Android - upravljanje memorijom

An abstract graphic on the left side of the slide, composed of several thick, curved lines. The lines are colored in a gradient from orange at the bottom to pink at the top. They form a shape that resembles a stylized letter 'A' or a large, open 'C' that is tilted to the right.

**Thank you for  
your attention!**