



Dretve

Procesi i dretve

Vlasništvo nad resursima

- Proces uključuje virtualni adresni prostor za sliku procesa
 - OS obavlja zaštitnu funkciju kako bi spriječio neželjene smetnje između procesa s obzirom na resurse

Planiranje/izvršenje

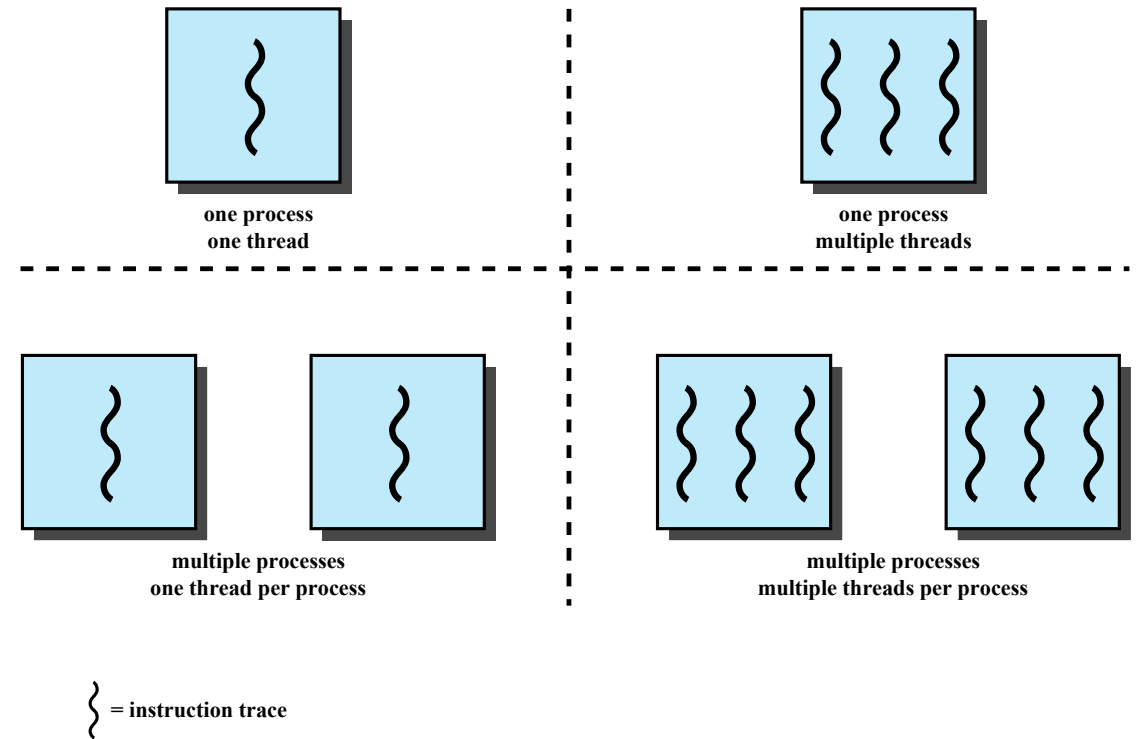
- Slijedi put izvršenja koji može biti isprepleten s drugim procesima
 - Proces ima stanje izvršenja (Running, Ready, itd.) i prioritet slanja

Procesi i dretve

- Jedinica za otpremu naziva se *dretva* ili *lagani process*
- Jedinica vlasništva nad resursom naziva se *procesom* ili *zadatkom*
- *Multithreading* - sposobnost OS-a da podrži više, istodobnih puteva izvršenja unutar jednog procesa

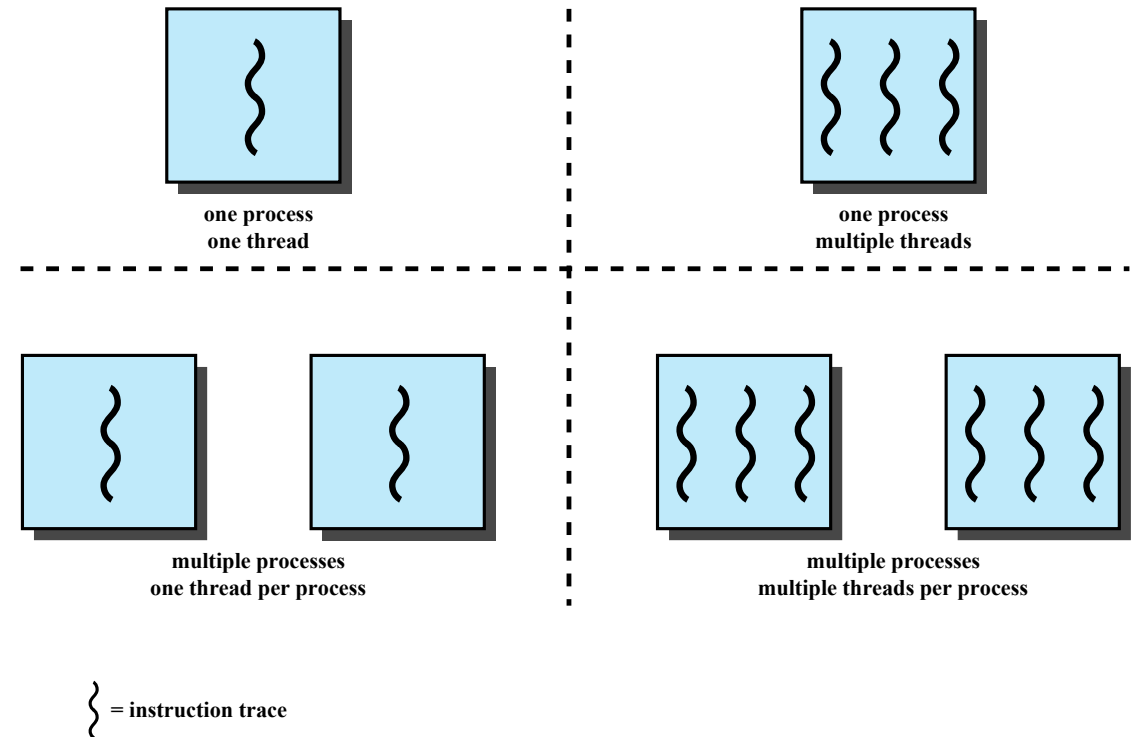
Jednodretveni pristupi

- Jednodretveno izvršenje po procesu, u kojoj se koncept dretve ne prepoznaje, naziva se pristupom s jednom dretvom
- MS-DOS je primjer



Višedretveni pristupi

- Desna polovica slike prikazuje višedretveni pristupe
- Java run-time okruženje je primjer sustava jednog procesa s više dretvi



Proces

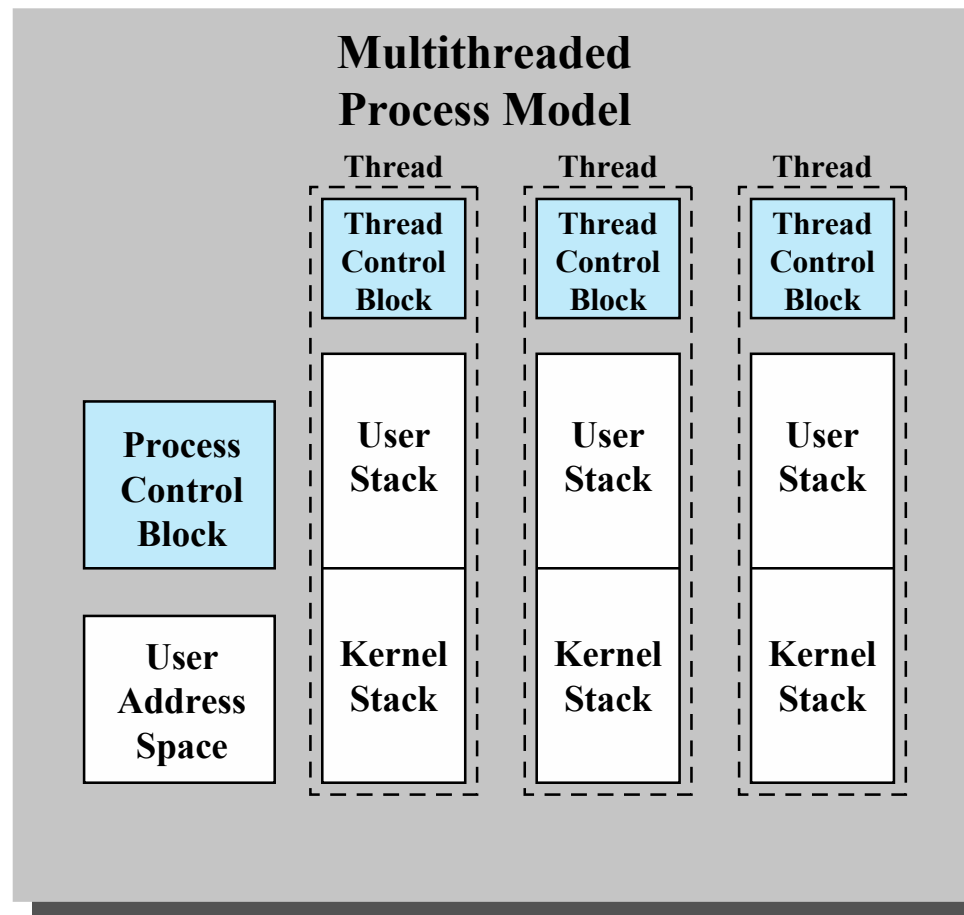
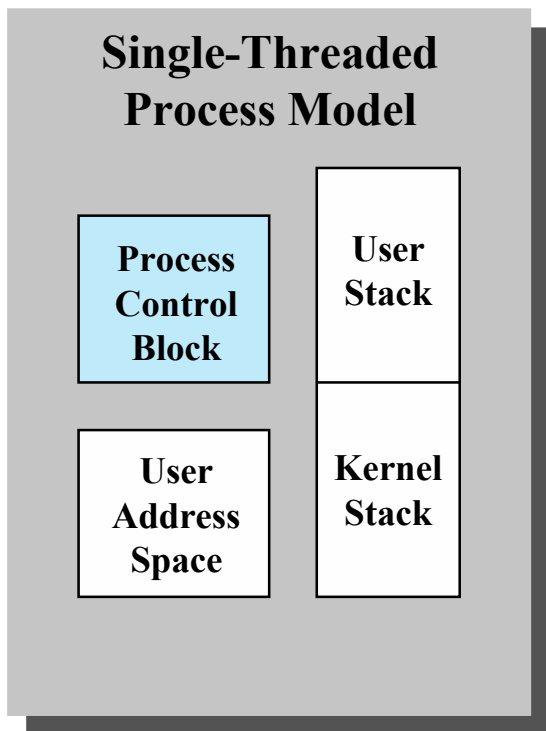
- Definiran u višedretvenom okruženju kao "jedinica dodjele resursa i jedinica zaštite,,
- Povezano s procesima:
 - Virtualni adresni prostor koji drži sliku procesa
 - Zaštićeni pristup:
 - Procesori
 - Ostali procesi (za međuprocesnu komunikaciju)
 - Datoteke
 - I/O resursi (uređaji i kanali)

Jedna ili više dretvi u procesu

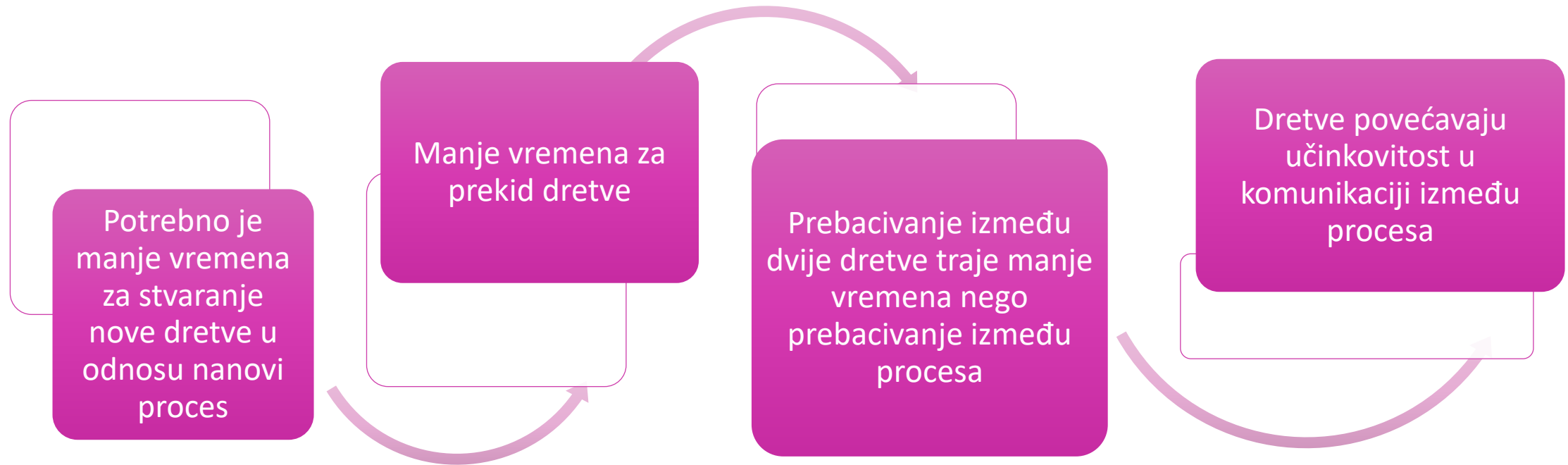
Svaka dretva ima:

- Stanje izvršenja (Pokrenuto, Spremno, itd.)
- Kontekst spremljene dretve kada se ne izvodi
- Izvršni stog
- Statičku pohranu po dretvi za lokalne varijable
- Pristup memoriji i resursima njegovih procesa, podijeljen sa svim ostalim dretvama u tom procesu

Jednodretveni vs višedretveni



Ključne prednosti dretvi



Korištenje dretvi u jednokorisničkim sustavima

- Rad u prvom planu i pozadini
- Asinkrona obrada
- Brzina izvršenja
- Modularna programska struktura

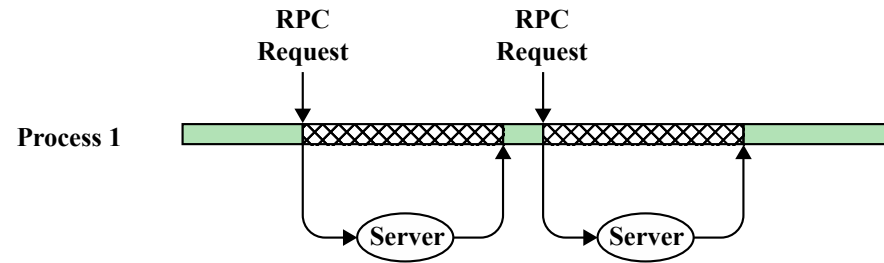
Dretve

- U OS-u koji podržava dretve, dodjeljivanje procesorskog vremena se obavljaju na bazi dretve
- Većina informacija o stanju koje se bave izvršenjem održava se u strukturama podataka na razini dretve
 - Suspendiranje procesa uključuje suspenziju svih dretvi procesa
 - Završetak procesa prekida sve dretve unutar procesa

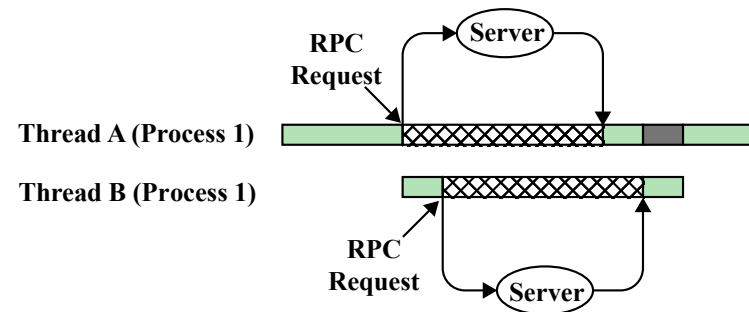
Stanja izvršenja dretve

- Ključna stanja za dretve su:
 - Running
 - Ready
 - Blocked
- Operacije dretvi povezane s promjenom stanja su:
 - Spawn
 - Block
 - Unblock
 - Finish




Remote Procedure Call korištenjem dretvi



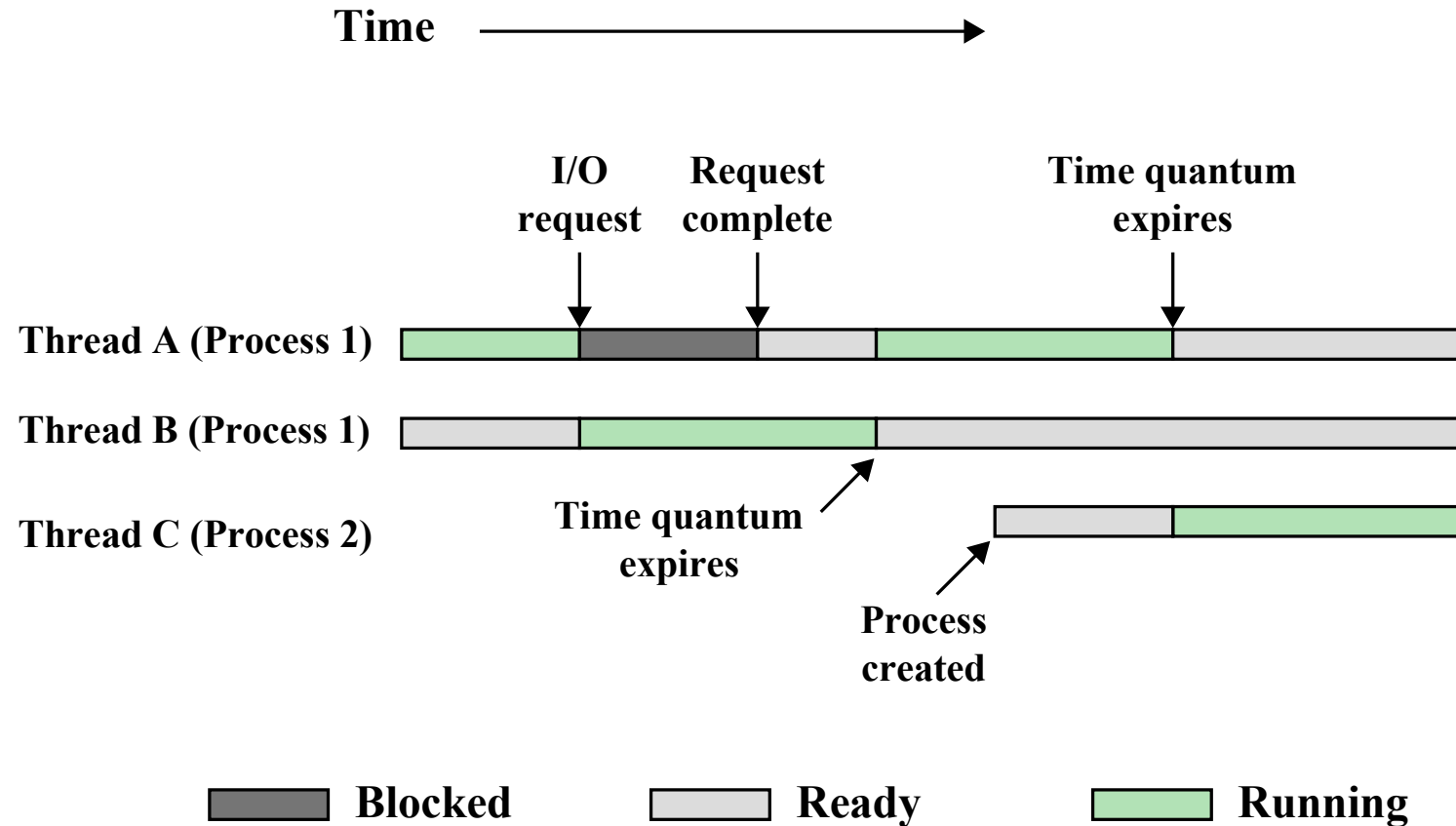
(a) RPC Using Single Thread



(b) RPC Using One Thread per Server (on a uniprocessor)

-  Blocked, waiting for response to RPC
-  Blocked, waiting for processor, which is in use by Thread B
-  Running

Multidretvenost na procesoru s jednom jezgrom



Sinkronizacija dretvi

- Potrebno je sinkronizirati aktivnosti različitih dretvi
 - Sve dretve procesa dijele isti adresni prostor i druge resurse
 - Svaka izmjena resursa od strane jedne dretve utječe na druge dretve u istom procesu

Vrste dretvi

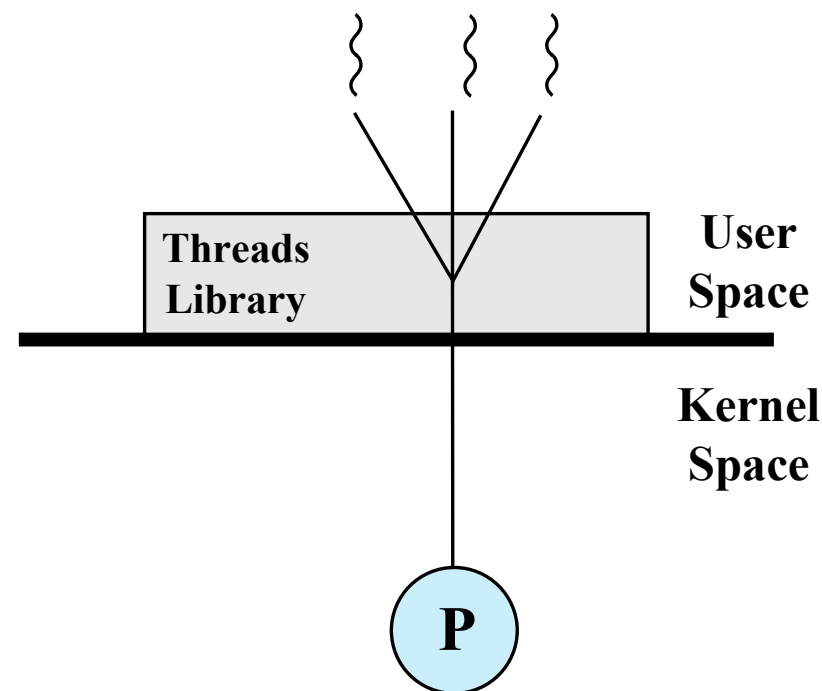


Korisničke
dretve (ULT)

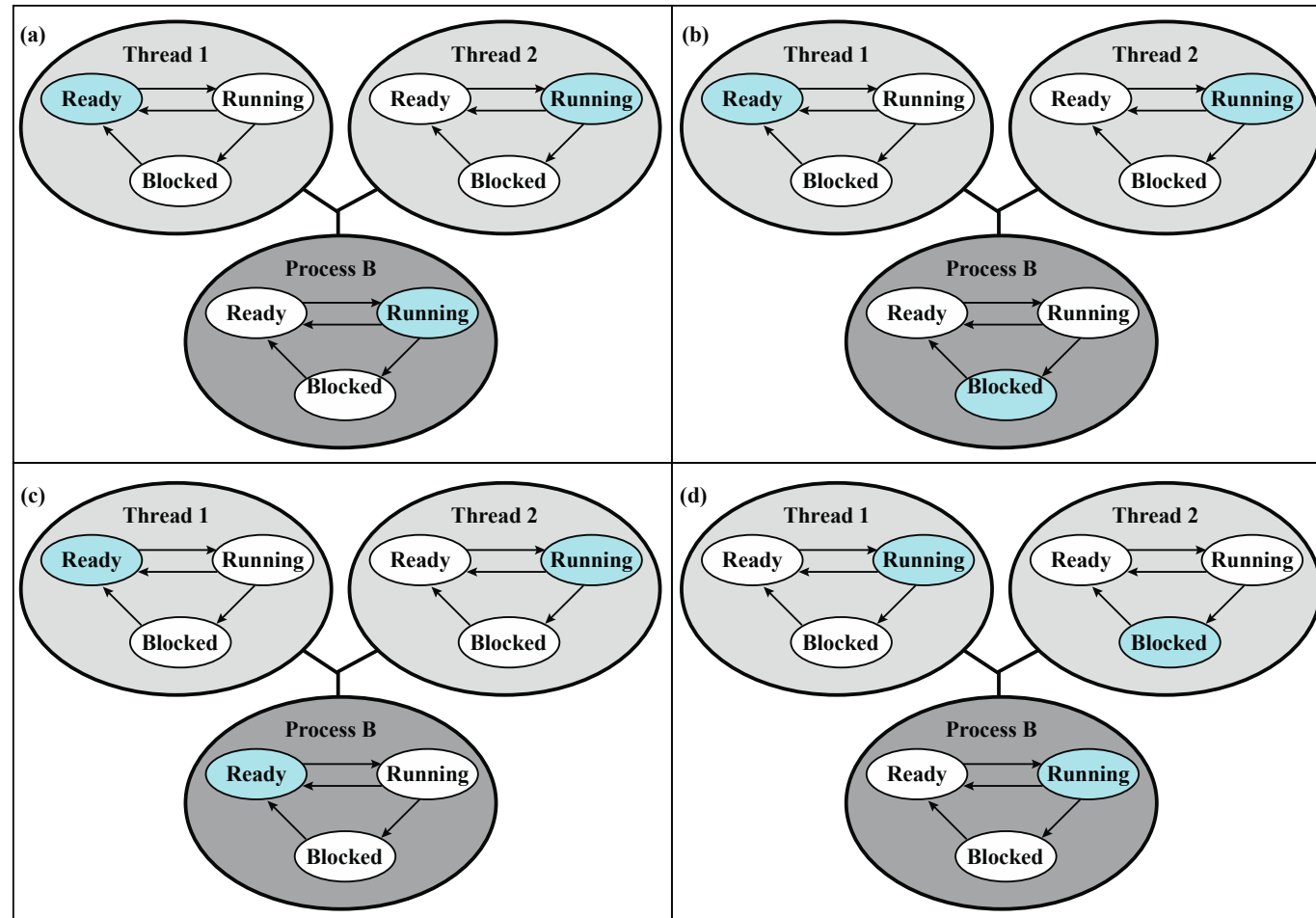
Jezgrine
dretve (KLT)

Korisničke dretve (ULT)

- Sve upravljanje dretvama obavlja aplikacija
- Jezgra nije svjestan postojanja dretvi

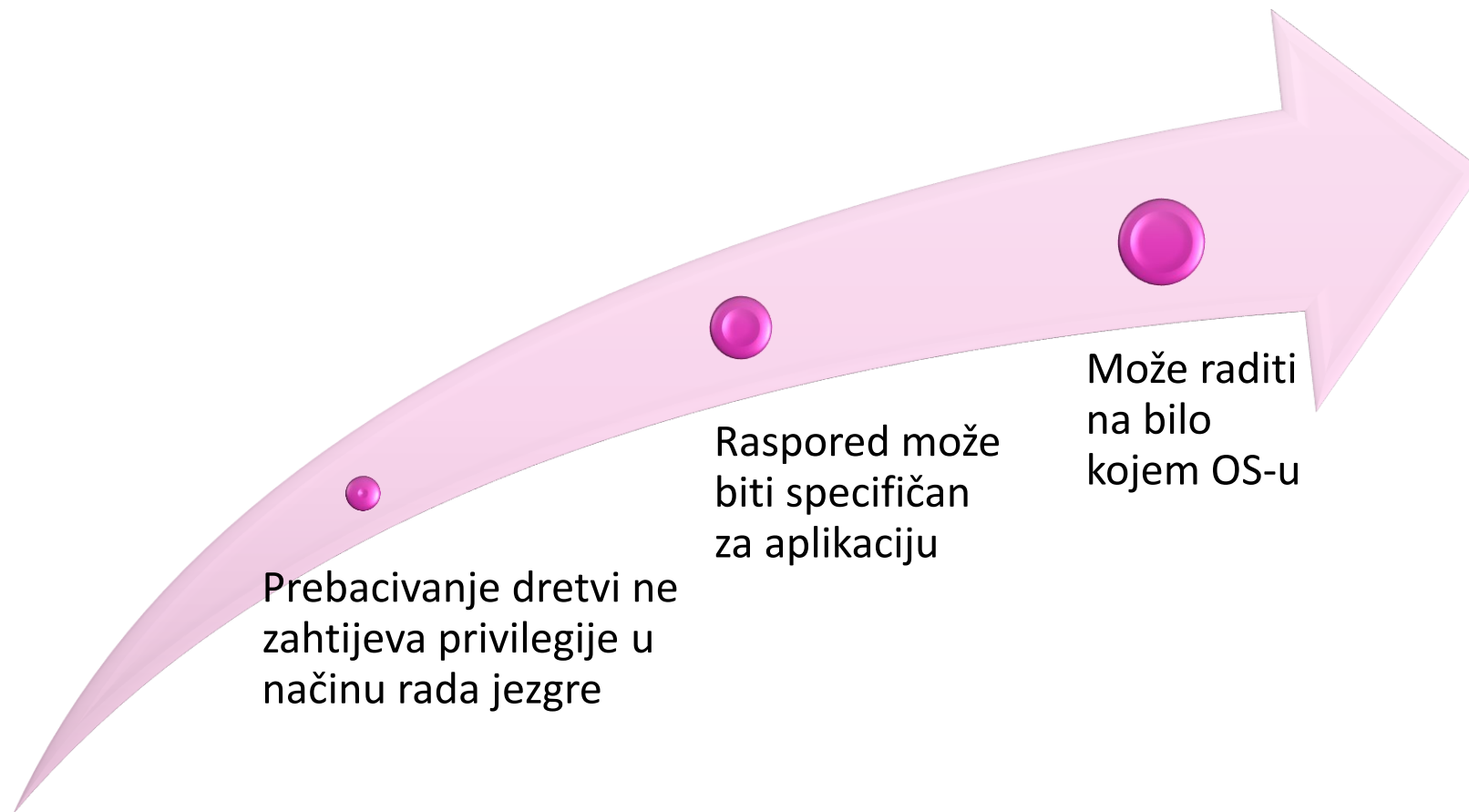


Statusi procega i dretve u korisničkom modu



Colored state
is current state

Prednosti korisničkih dretvi



Nedostaci korisničkih dretvi

- U tipičnom OS-u mnogi sistemski pozivi su blokirani
 - Kao rezultat toga, kada ULT izvrši poziv sustava, ne samo da je ta dretva blokirana, već su blokirane i sve dretve unutar procesa
- U čistoj ULT strategiji, višedretvena aplikacija ne može iskoristiti prednosti višeproceniranja
 - Kernel dodjeljuje jedan proces samo jednom procesoru u isto vrijeme, dakle, samo jedna dretva unutar procesa može se izvršavati istovremeno

Prevladavanje nedostataka ULT-a

Jacketing - oblaganje

- Svrha je pretvoriti blokirajući sistemski poziv u neblokirajući sistemski poziv

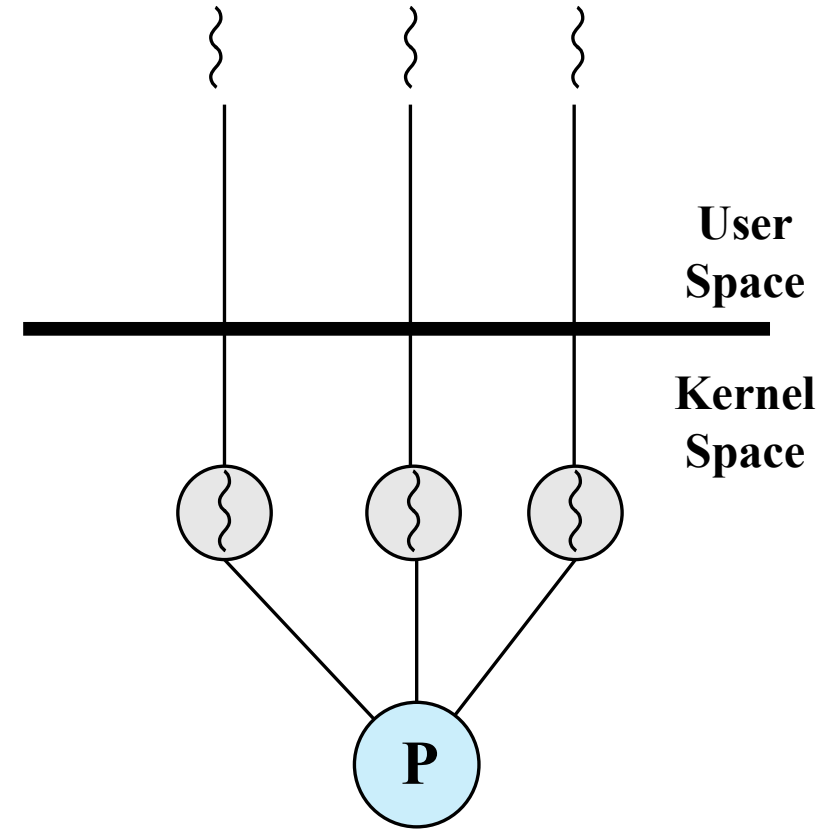


Pisanje aplikacije kao više procesa, a ne više dretvi

- Ovaj pristup eliminira glavnu prednost dretvi

Dretve jezgre (KLTs)

- Upravljanje dretvama obavlja jezgra
 - Ne postoji kod za upravljanje dretvama na razini aplikacije, samo sučelje za programiranje aplikacije (API) za podršku dretvama
 - Windows je primjer ovakvog pristupa



Prednosti KLT-a

- Kernel može istovremeno rasporediti više dretvi iz istog procesa na više procesora
- Ako je jedna dretva u procesu blokirana, kernel može izvršavati drugu dretve istog procesa
- Same rutine kernela mogu biti višedretvene

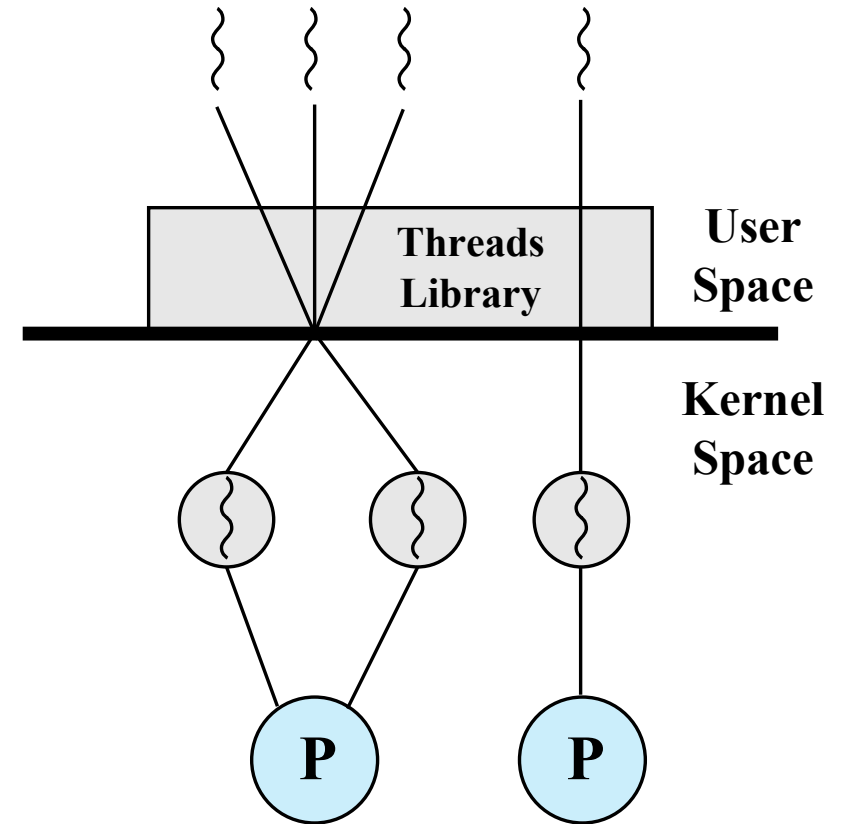
Nedostaci KLT-a

- Prijenos kontrole s jedne niti na drugu unutar istog procesa zahtijeva prebacivanje na kernel načina rada

Operacija	ULT	KLT	Procesi
Null Fork	34 μ s	948 μ s	11300
Signal Wait	37 μ s	441 μ s	1,840

Kombinirani pristup

- Kreiranje dretvi se obavlja u potpunosti u korisničkom prostoru, kao i najveći dio rasporeda i sinkronizacije unutar aplikacije
- Solaris je dobar primjer



Odnos između dretvi i procesa

Ops	Description	Primjer OS-a
1:1	Svaka dretva izvršenja je jedinstveni proces s vlastitim adresnim prostorom i resursima.	Traditional UNIX implementations
M:1	Proces definira adresni prostor i dinamičko vlasništvo resursa. U tom procesu može se stvoriti i izvršiti više dretvi.	Windows NT, Solaris, Linux, OS/2, OS/390, MACH
1:M	Dretva može migrirati iz jednog procesnog okruženja u drugo. To omogućuje lako premještanje dretvi između različitih sustava.	Ra (Clouds), Emerald
M:N	Kombinira M:1 i 1:M slučajeve.	TRIX

Aplikacije koje imaju koristi

- Višedretvene izvorne aplikacije
 - Karakterizira ih mali broj procesa s velikim brojem dretvi
- Višeprocесne aplikacije
 - Karakterizira ih prisutnost više jednodretvenih procesa
- Java aplikacije
 - Sve aplikacije koje koriste Java 2 Platform, Enterprise Edition mogu imati koristi od višejezgrene tehnologije
- Aplikacije s više instanci
 - Više instanci aplikacije paralelno

Upravljanje procesima i dretvama (Windows)

- **Aplikacija** se sastoji od jednog ili više procesa
- Svaki **proces** pruža resurse potrebne za izvršavanje programa
- **Dretva** je entitet unutar procesa koji se može rasporediti za izvršenje
- **Objekt posla** omogućuje upravljanje grupama procesa kao jedinicom
- **Skup dretvi** je skup radničkih dretvi koje učinkovito izvršavaju asinkrone povratne pozive u ime aplikacije
- **Nit** je jedinica izvršenja koju aplikacija mora ručno zakazati
- **Zakazivanje u korisničkom načinu rada (UMS)** je lagani mehanizam koji aplikacije mogu koristiti za zakazivanje vlastitih dretvi

Upravljanje pozadinskim zadacima i životnim ciklusima aplikacija

- Počevši od Windowsa 8 programeri su odgovorni za upravljanje stanjem svojih pojedinačnih aplikacija
- Prethodne verzije sustava Windows davale su korisniku potpunu kontrolu nad životnim vijekom procesa
- U novom Metro sučelju Windows preuzima procesni životni ciklus aplikacije
 - Ograničen broj aplikacija može se izvoditi uz glavnu aplikaciju u korisničkom sučelju Metro pomoću funkcije SnapView
 - Istovremeno se može pokrenuti samo jedna aplikacija iz Store-a
- Live Tiles daju izgled aplikacija koje stalno rade na sustavu
 - U stvarnosti primaju push obavijesti i ne koriste resurse sustava za prikaz ponuđenog dinamičkog sadržaja

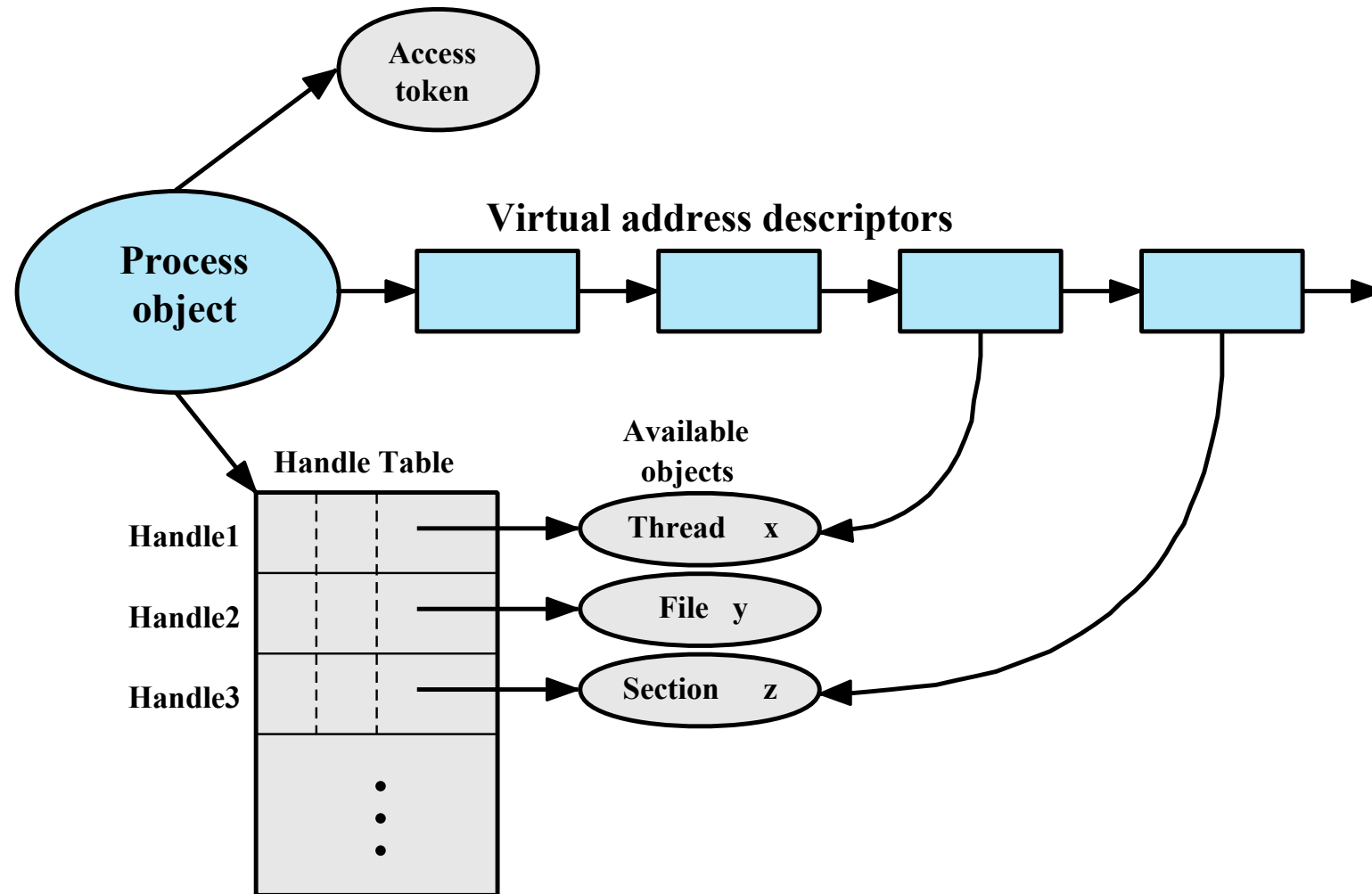
Metro sučelje

- Aplikacija u fokusu kod Metro sučelju ima pristup svim resursima procesora, mreže i diska koji su dostupni korisniku
 - Sve ostale aplikacije su obustavljene i nemaju pristup tim resursima
- Kada aplikacija uđe u obustavljeni način rada, trebao bi se pokrenuti događaj za pohranu stanja podataka korisnika
 - To je odgovornost programera aplikacije
- Windows može prekinuti pozadinsku aplikaciju
 - Morate spremiti stanje svoje aplikacije kada je suspendirana, u slučaju da je Windows prekine kako biste kasnije mogli vratiti njeno stanje
 - Kada se aplikacija vrati u fokus pokreće se drugi događaj za učitavanje stanja iz memorije

Procesi (Windows)

- Važne karakteristike Windows procesa su:
 - Windows procesi se implementiraju kao Objekti
 - Proces se može kreirati kao novi proces ili kopija postojećeg procesa
 - Izvršni proces može sadržavati jednu ili više dretvi
 - Procesi i dretve imaju ugrađene mogućnosti sinkronizacije

Procesi i resursi



Objekti procesa i dretvi

- Windows koristi dvije vrste objekata povezanih s procesom:

Procesi

- Entitet koji odgovara korisničkom poslu ili aplikaciji koja posjeduje resurse

Dretve

- Jedinica rada koja se može raspoređivati za izvršavanje sekvencijalno i koja podržava prekidanje

Atributi procesa - Windows

Process ID	A unique value that identifies the process to the operating system.
Security descriptor	Describes who created an object, who can gain access to or use the object, and who is denied access to the object.
Base priority	A baseline execution priority for the process's threads.
Default processor affinity	The default set of processors on which the process's threads can run.
Quota limits	The maximum amount of paged and nonpaged system memory, paging file space, and processor time a user's processes can use.
Execution time	The total amount of time all threads in the process have executed.
I/O counters	Variables that record the number and type of I/O operations that the process's threads have performed.
VM operation counters	Variables that record the number and types of virtual memory operations that the process's threads have performed.
Exception/debugging ports	Interprocess communication channels to which the process manager sends a message when one of the process's threads causes an exception. Normally, these are connected to environment subsystem and debugger processes, respectively.
Exit status	The reason for a process's termination.

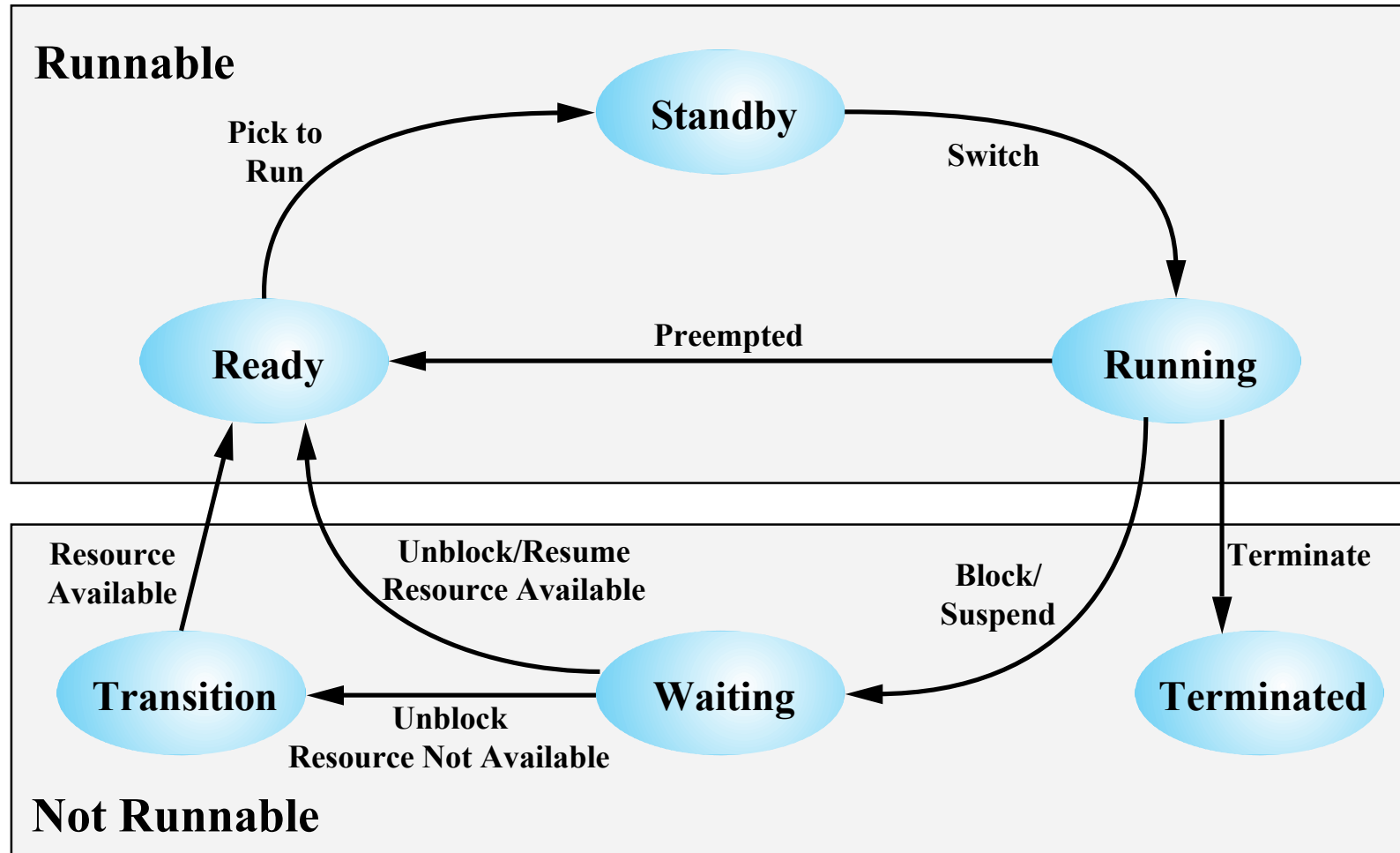
Atributi dretve - Windows

Thread ID	A unique value that identifies a thread when it calls a server.
Thread context	The set of register values and other volatile data that defines the execution state of a thread.
Dynamic priority	The thread's execution priority at any given moment.
Base priority	The lower limit of the thread's dynamic priority.
Thread processor affinity	The set of processors on which the thread can run, which is a subset or all of the processor affinity of the thread's process.
Thread execution time	The cumulative amount of time a thread has executed in user mode and in kernel mode.
Alert status	A flag that indicates whether a waiting thread may execute an asynchronous procedure call.
Suspension count	The number of times the thread's execution has been suspended without being resumed.
Impersonation token	A temporary access token allowing a thread to perform operations on behalf of another process (used by subsystems).
Termination port	An interprocess communication channel to which the process manager sends a message when the thread terminates (used by subsystems).

Višedretvenost



Stanja dretvi



Procesi - Solaris

- Koristi četiri koncepta povezana s dretvama:

Proces

- Uključuje korisnikov adresni prostor, stog i kontrolni blok procesa

User-level Threads

- Jedinica izvršenja koju je kreirao korisnik unutar procesa

Lightweight Processes (LWP)

- Mapiranje između ULT-ova i dretvi kernela

Kernel Threads

- Temeljni entiteti koji se mogu rasporediti i dodjeliti na izvršavanje na jednom od procesora sustava

Struktura podataka laganog procesa (LWP) uključuje:

- LWP identifikator
- Prioritet LWP-a i dretvu kernela koja ga podržava
- Maska signala koja govori kernelu koji će signali biti prihvaćeni
- Spremljene vrijednosti registara na razini korisnika
- Stog kernela za ovaj LWP, koji uključuje argumente sistemskog poziva, rezultate i kodove pogrešaka za svaku razinu poziva
- Podaci o korištenju resursa i profiliranju
- Pokazivač na odgovarajuću dretvu kernela
- Pokazivač na strukturu procesa

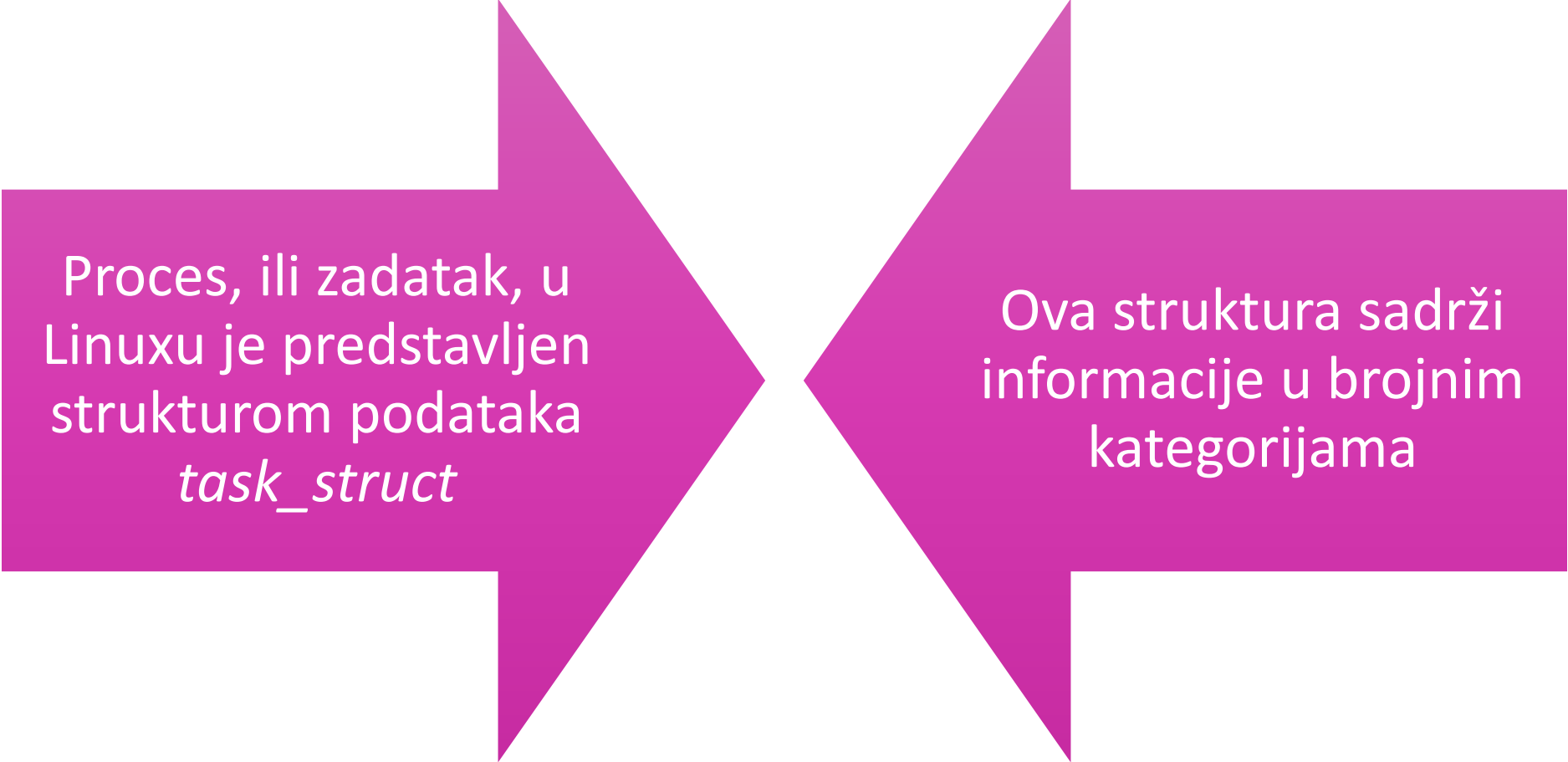
Prekidi kao dretve

- Većina operativnih sustava sadrži dva temeljna oblika istodobne aktivnosti:
 - Procesi (dretve): međusobno surađuju i upravljaju korištenjem zajedničkih struktura podataka koji provode međusobno isključivanje i sinkroniziraju njihovo izvršavanje
 - Prekidi: sinkronizirani sprječavanjem njihovog rukovanja na određeno vrijeme
- Solaris objedinjuje ova dva koncepta u jedan model, naime dretve kernela, i mehanizme za raspoređivanje i izvršavanje dretvi kernela
 - Da biste to radilo, prekidi se pretvaraju u dretve kernela

Rješenje - Solaris

- Solaris koristi skup dretvi kernela za rukovanje prekidima
 - Prekidna dretva ima svoj identifikator, prioritet, kontekst i stog
 - Jezgra kontrolira pristup strukturama podataka i sinkronizira se među prekidnim nitima koristeći objekte za međusobno isključivanje
 - Dretve prekida imaju veći prioritet od svih ostalih vrsta dretvi kernela

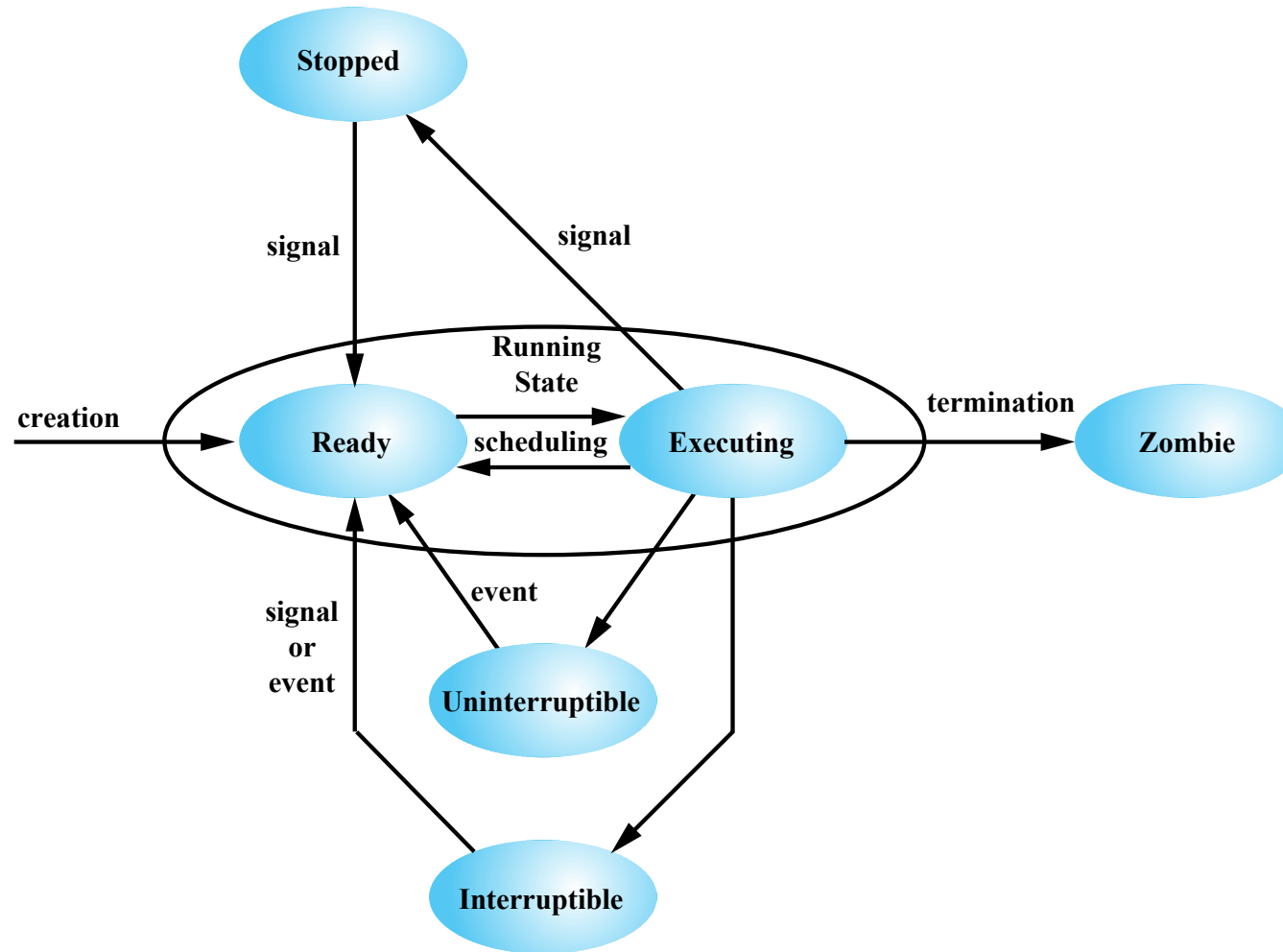
Linux zadaci



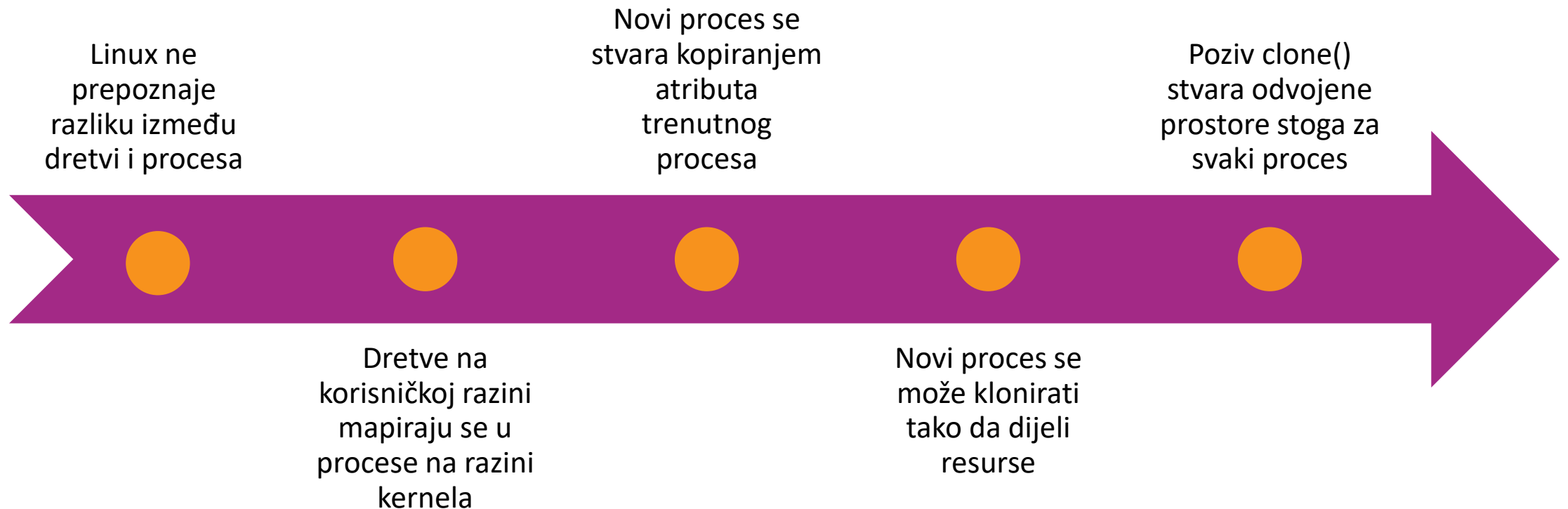
Proces, ili zadatak, u Linuxu je predstavljen strukturom podataka *task_struct*

Ova struktura sadrži informacije u brojnim kategorijama

Stanja Procesa/Dretvi - Linux



Dretve – Linux



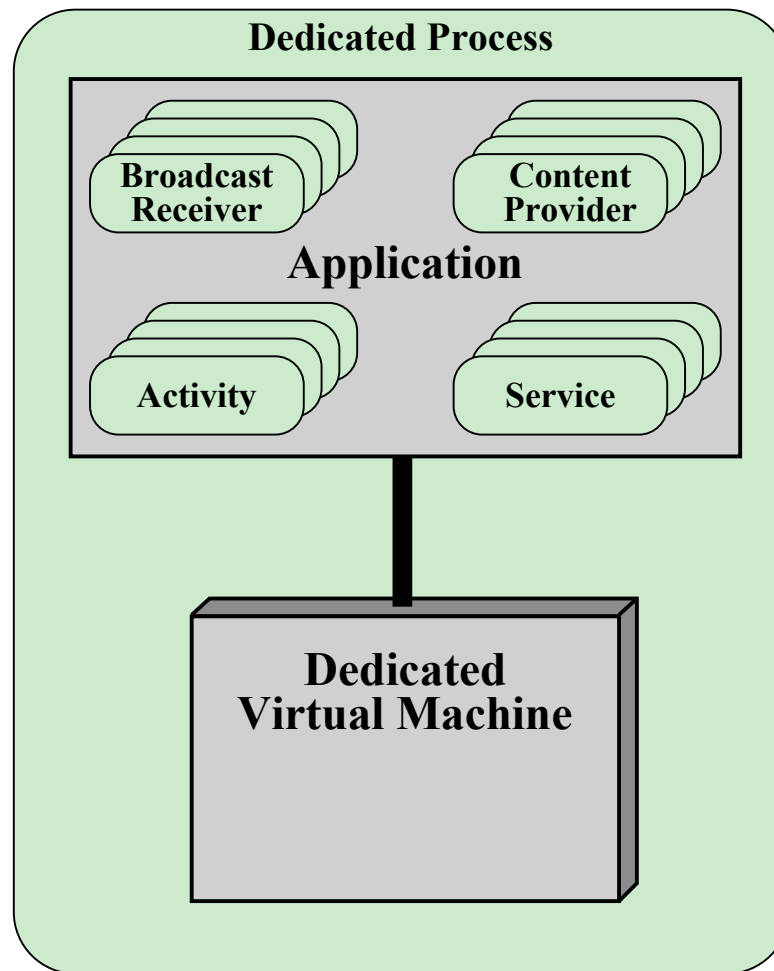
Imenski prostor - Linux

- Imenski prostor (namespace) omogućuje procesu da ima drugačiji pogled na sustav od ostalih procesa koji imaju druge imenske prostore
- Trenutno postoji šest imenskih prostora u Linuxu
 - mnt
 - pid
 - net
 - ipc
 - uts
 - user

Upravljanje procesima i dretvama - Android

- Svaka Android aplikacija sastoji se od jedne ili više instanci jedne ili više od četiri vrste komponenti aplikacije
- Svaka komponenta ima posebnu ulogu u cjelokupnom ponašanju aplikacije, a svaka komponenta može se aktivirati neovisno unutar aplikacije, pa čak i drugim aplikacijama
- Četiri vrste komponenti:
 - Aktivnosti
 - Usluge
 - Pružatelji sadržaja
 - Prijemnici za emitiranje

Android aplikacije

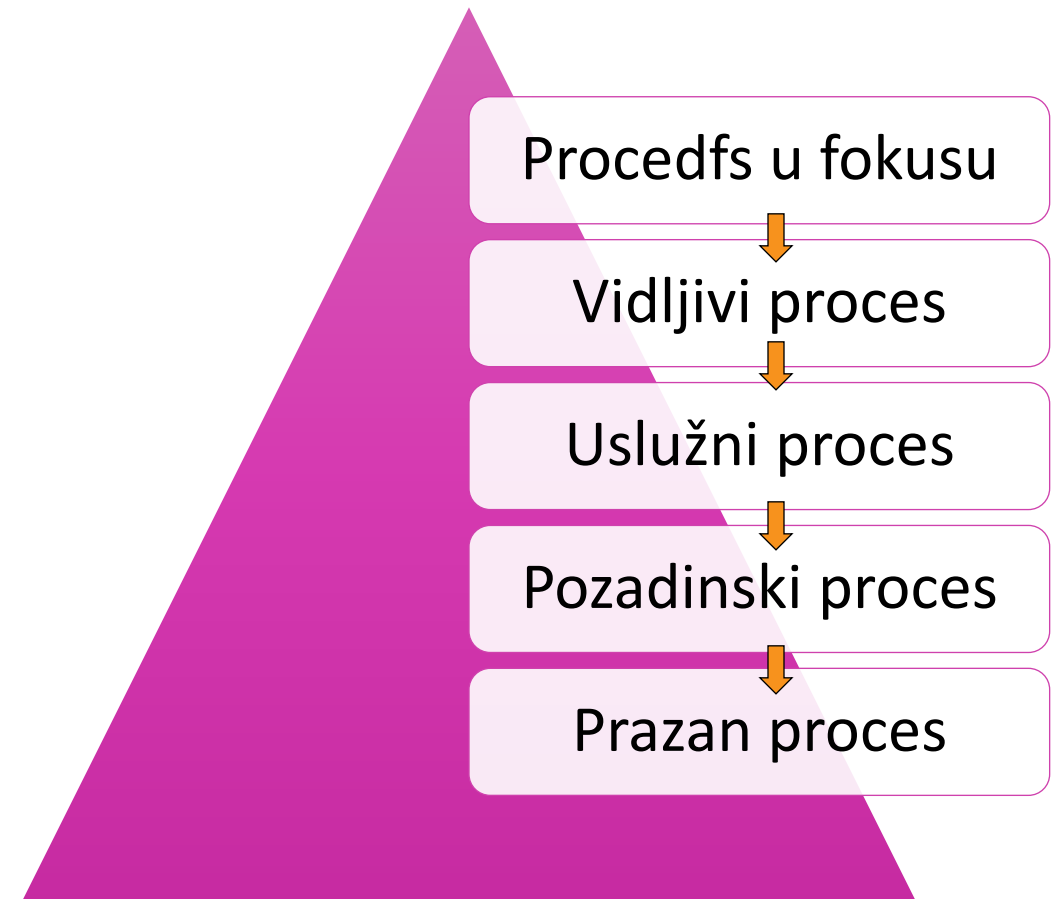


Aktivnosti - Android

- Aktivnost je komponenta aplikacije koja pruža zaslon s kojim korisnici mogu komunicirati kako bi nešto učinili
- Svaka aktivnost ima prozor u kojem crta svoje korisničko sučelje
- Prozor obično ispunjava zaslon, ali može biti manji od zaslona i lebđeti iznad drugih prozora
- Aplikacija može uključivati više aktivnosti
- Kada je aplikacija pokrenuta, jedna aktivnost je u prvom planu, a ona je ta koja je u interakciji s korisnikom
- Aktivnosti su raspoređene u LIFO stog redoslijedom kojim se otvara svaka aktivnost
- Ako se korisnik prebaci na neku drugu aktivnost unutar aplikacije, nova aktivnost se kreira i gura na vrh stoga, dok prethodna aktivnost u prvom planu postaje druga stavka na stogu za ovu aplikaciju

Procesi i dretve

- Hijerarhija prioriteta koristi se za određivanje koji proces ili procese treba ubiti kako bi se povratili potrebni resursi
- Procesi se uništavaju počevši od najnižeg prioriteta
- Razine hijerarhije, u padajućem redoslijedu prvenstva su:



Grand Central Dispatch (GCD) – OS X

- Pruža skup dostupnih dretvi
- Dizajneri mogu odrediti dijelove aplikacija, zvane blokovi, koji se mogu slati neovisno i izvoditi istodobno
- Istodobnost se temelji na broju dostupnih jezgri i kapacitetu dretvi sustava

Blok

- Jednostavno proširenje jezika
- Blok definira samostalnu jedinicu rada
- Omogućuje programeru da inkapsulira složene funkcije
- Raspoređeno i otpremljeno u redovima
- Šalje se po principu FIFO
- Može se povezati s izvorom događaja, kao što je mjerač vremena, mrežni soket ili deskriptor datoteke

Sažetak

- Procesi i dretve
 - Višedretvenost
 - Funkcionalnost dretvi
- Vrste dretvi
 - Dretve na razini korisnika i kernela
- Višejezgrenost i višedretvenost
 - Izvršavanje programa na više jezgre
- Upravljanje procesima i dretvama - Windows
 - Upravljanje pozadinskim zadacima i životnim ciklusima aplikacija
 - Windows process
 - Objekti procesa i dretvi
 - Višedretvenost
 - Stanja dretvi
 - Podrška za podsustave OS-a
- Solaris dretve i SMP upravljanje
 - Višedretvena arhitektura
 - Motivacija
 - Struktura procesa
 - Izvođenje dretvi
 - Prekida kao dretve
- Upravljanje procesima i dretvama na Linux-u
 - Zadaci/dretve/imenski prostori
- Android proces i upravljanje dretvama
 - Android aplikacije
 - Aktivnosti
 - Procesi i dretve
- Mac OS X grand central dispatch



**Thank you for
your attention!**