

OPERATIVNI SISTEMI

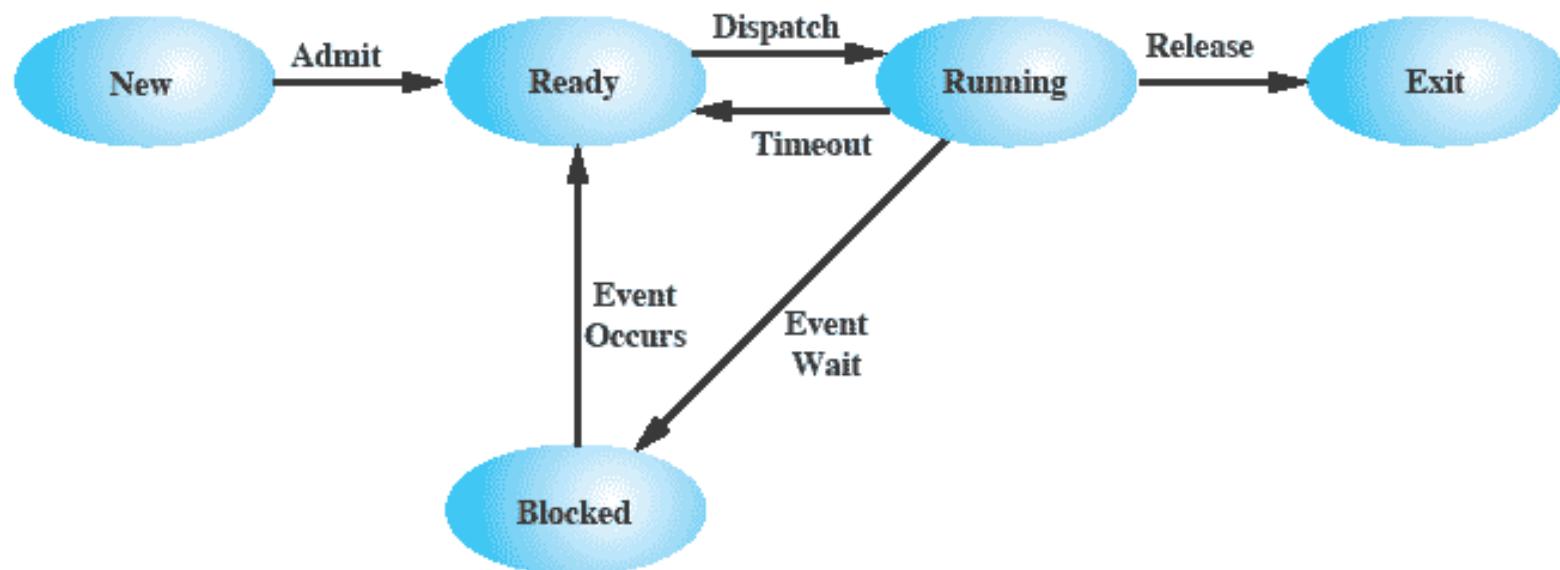
Slajdovi su kreirani na osnovu knjige "Operativni sistemi, principi unutrašnje organizacije i dizajna, 7. izdanje", William Stallings, CET, Beograd, 2013.

Niti

Procesi

- Program u izvršavanju
- Proces kao jedinica izvršavanja i raspoređivanja
 - procesor izvršava instrukcije procesa
 - isprepleteno sa izvršavanjem drugih procesa
 - OS raspoređuje procese
- Proces kao jedinica kojoj se dodeljuje resurs
 - proces sadrži virtuelni adresni prostor u koji je smeštena slika procesa
 - procesu se može dodeliti vlasništvo nad memorijom, U/I resursom ili fajlom

Stanja procesa

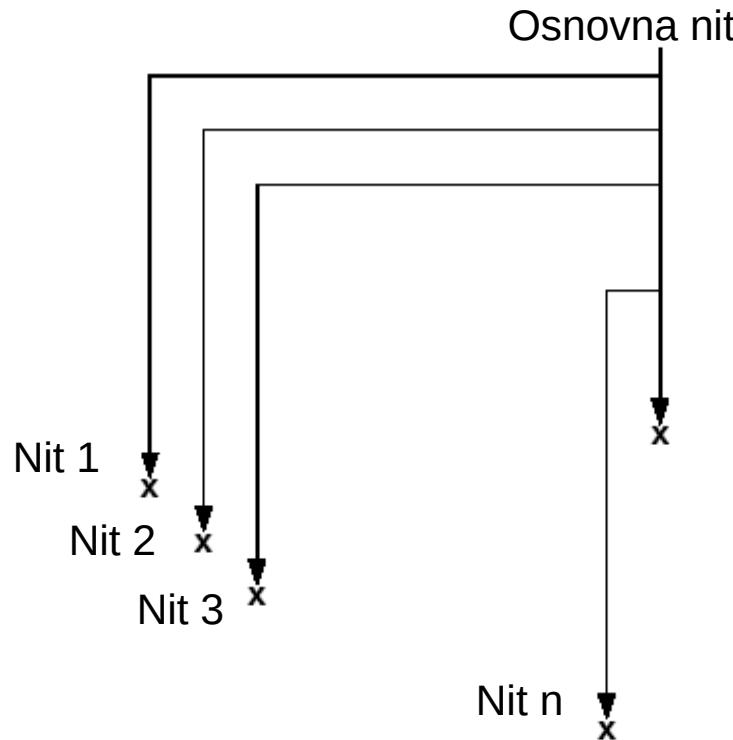


Stanja procesa

- Izvršavanje
 - proces čije instrukcije procesor trenutno izvršava
- Spreman
 - proces koji je spreman za izvršavanje, ali trenutno procesor ne izvršava njegove instrukcije
- Blokiran (U čekanju)
 - proces koji se ne može izvršavati dok se ne pojavi neki događaj
 - npr. završetak U/I operacije
- Novi
 - proces koji je upravo stvoren, ali ga OS još nije prihvatio u red spremnih procesa
- Izlaz
 - proces koji je OS uklonio iz reda spremnih procesa zato što je završio rad

Višenitna obrada

- Sposobnost OS da podrži više konkurentnih putanja izvršavanja jednog procesa
- Nit je jedan tok izvršavanja u okviru procesa



Višenitna obrada



Paralelizam





Niti u C++11 standardu

- Predstavljene objektom klase `thread` iz zaglavlja `<thread>`
- Instanciranje objekta kreira novu nit koja postaje spremna
- Parametar konstruktora klase `thread` je funkcija koju nit izvršava
 - Ova funkcija se zove telo niti
- Ulazna tačka programa je i dalje `main` funkcija
- Završetak je kad se završi `main`

Primer višenitne aplikacije

- Primeri/Niti/PrimerNiti

Odnos niti stvaralac prema stvorenoj niti

- Metoda join() klase thread
 - Stvaralac čeka da stvorena nit završi rad
 - Time se spajaju dva toka izvršavanja
 - Metoda je **blokirajuća**
 - Stvaralac će biti blokiran dok stvorena nit ne završi rad
- Metoda detach() klase thread
 - Razdvaja stvaraoca od stvorene niti
 - Tada je dozvoljeno da stvaralac završi rad bez obzira na rad stvorene niti

Prenos parametara u nit

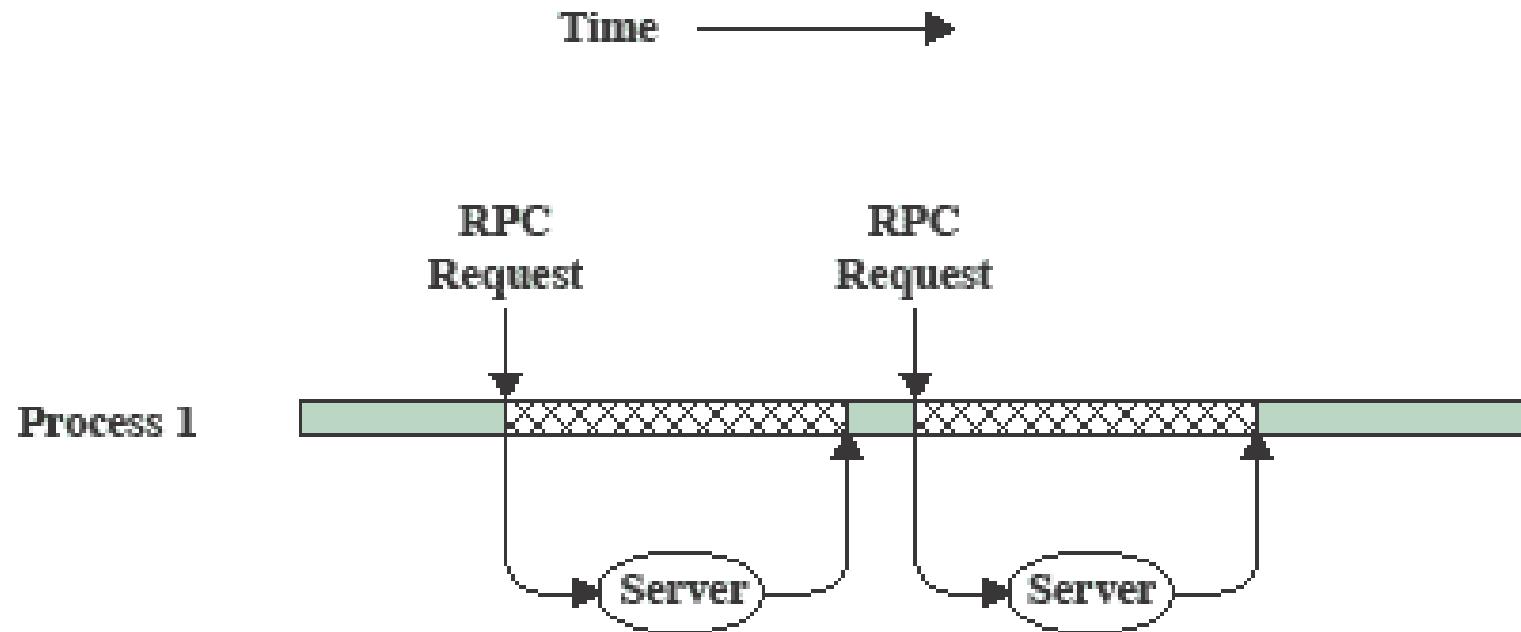
- Parametri tela niti se prenose kao parametri konstruktora klase thread nakon naziva funkcije
- Povratna vrednost funkcije se zanemaruje
- Potrebno je prenositi objekte po referenci u funkciju
- Mora se eksplicitno preneti referenca funkcijom **ref(objekat)**

Primer prenosa parametara u nit

- Primeri/Niti/ParametriNiti

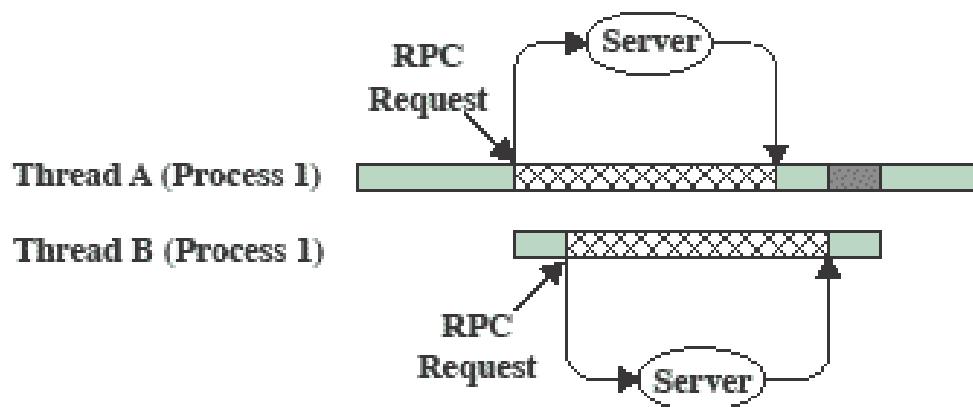
Primer poboljšanja performansi upotrebom višenitne obrade

- Program upućuje dva zahteva udaljenom serveru da bi izračunao rezultat
- Varijanta sa jednom niti



Primer poboljšanja performansi upotrebom višenitne obrade

- Varijanta sa dve niti
- Blokiranje jedne niti ne blokira drugu nit



Koristi od višenitne obrade

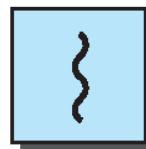
- Pozadinski posao
 - dugačke ili blokirajuće pozadinske operacije se mogu izvršavati u posebnoj niti, npr. upis u log fajl
 - interakcija sa korisnikom je moguća i dok druga nit nije završila operaciju
- Asinhrona obrada
 - Naredba predaje posao drugoj niti
 - Nit koja je predala posao može da nastavi izvršavanje drugih naredbi dok se prethodna naredba još nije izvršila
- Brzina izvršavanja
 - dok je jedna nit procesa blokirana, druge mogu da se izvršavaju

Koristi od višenitne obrade

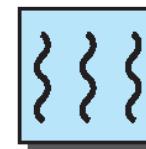
- Deljenje resursa unutar aplikacije
 - niti istog procesa dele istu memoriju
 - brža i jednostavnija komunikacija između niti nego između procesa
- Ekonomičnost
 - brže je napraviti novu nit nego poseban proces
 - brže je komutiranje između dve niti istog procesa nego između dva procesa
- Iskorišćenje višeprocesorske arhitekture
 - mogućnost ubrzanja aplikacije ukoliko se niti izvršavaju na različitim procesorima
- Modularna struktura programa
 - organizacija koda u više niti razdvaja logičke delove programa
 - jednostavnija implementacija i održavanje

Odnos procesa i niti

OS podržava samo jedan proces koji sadrži jednu nit (MS DOS)



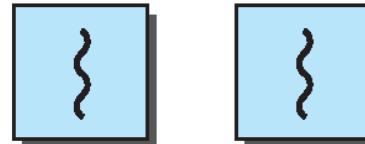
one process
one thread



one process
multiple threads

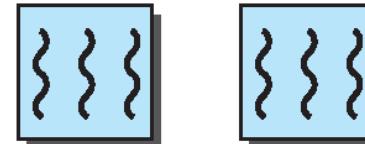
Okruženje sadrži jedan proces u kojem može postojati više niti (Java VM)

OS podržava više procesa od kojih svaki sadrži jednu nit (tradicionalni UNIX)



multiple processes
one thread per process

{ = instruction trace



multiple processes
multiple threads per process

OS podržava više procesa od kojih svaki podržava više niti (Windows, Linux, Solaris)

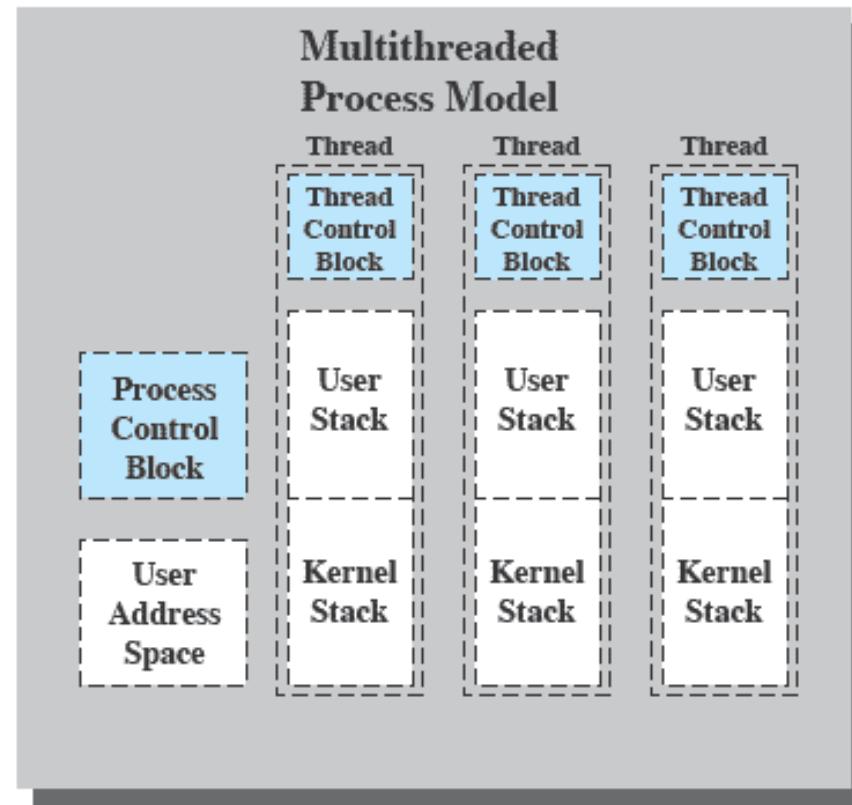
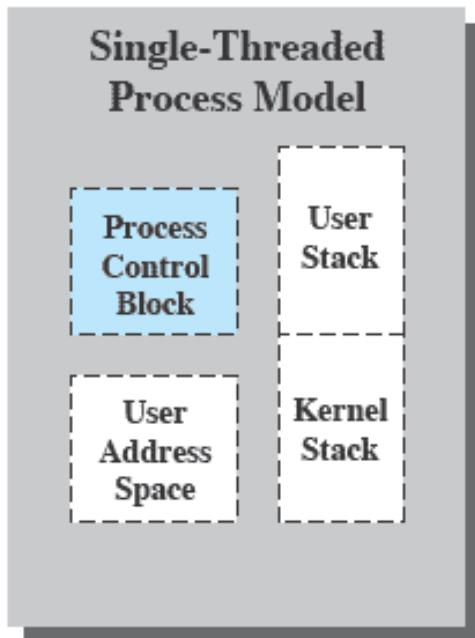
Proces u višenitnoj obradi

- Jedinica dodele resursa
 - virtuelni adresni prostor u kome je smeštena slika procesa
- Jedinica zaštite resursa
 - zaštićen pristup
 - procesoru
 - drugim procesima (međuprocesna komunikacija)
 - fajlovima
 - U/I resursima (uređaji, kanali)

Nit u višenitnoj obradi

- Jedan proces može da sadrži više niti
- Svaka nit ima
 - stanje izvršavanja
 - (izvršavanje, spremna, ...)
 - kontekst niti
 - sačuvano stanje procesorskih registara dok se ne izvršava
 - stek izvršavanja niti
 - pristup memoriji i resursima svog procesa
 - deljeno sa drugim nitima istog procesa

Model procesa sa višenitnom obradom



- Svaka nit ima svoj upravljački blok i stek
- Sve niti dele adresni prostor i imaju pristup istim podacima

Raspoređivanje niti

- Ako OS podržava niti, raspoređivanje se vrši na nivou niti
- Većina informacija o stanju izvršavanja čuva se na nivou niti
- Neke akcije utiču na sve niti
- OS tada mora raspoređivanje vršiti na nivou procesa
 - Npr. prekidanjem procesa prekidaju se sve njegove niti

Stanja niti

- Kao i kod procesa
 - Izvršavanje
 - Spreman
 - Blokiran
- Akcije kojima se menja stanje niti
 - Kreiranje
 - Kreira se upravljački blok niti i stek i nit se postavlja u stanje spremjan
 - Blokiranje
 - Kada nit mora da čeka na događaj
 - Sačuva se sadržaj procesorskih registara i procesor se prebacuje na neku drugu nit iz skupa spremnih niti
 - Deblokiranje
 - Kada se događaj koji je nit čekala desi
 - Završavanje

Sinhronizacija niti

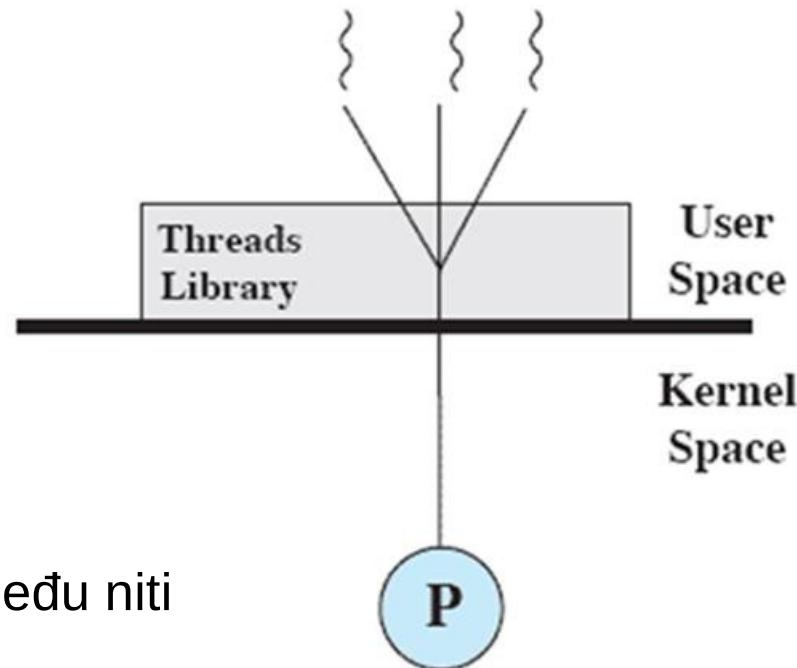
- Sve niti jednog procesa dele isti adresni prostor
- Potrebno je sinhronizovati pristup ovim deljenim resursima
- Bez sinhronizacije
 - može doći do neispravnog rada programa
 - resursi mogu doći u nekonzistentno stanje

Tipovi niti

- Niti nivoa korisnika
 - *User level threads* (ULT)
- Niti nivoa kernela
 - *Kernel level threads* (KLT)

Niti nivoa korisnika (ULT)

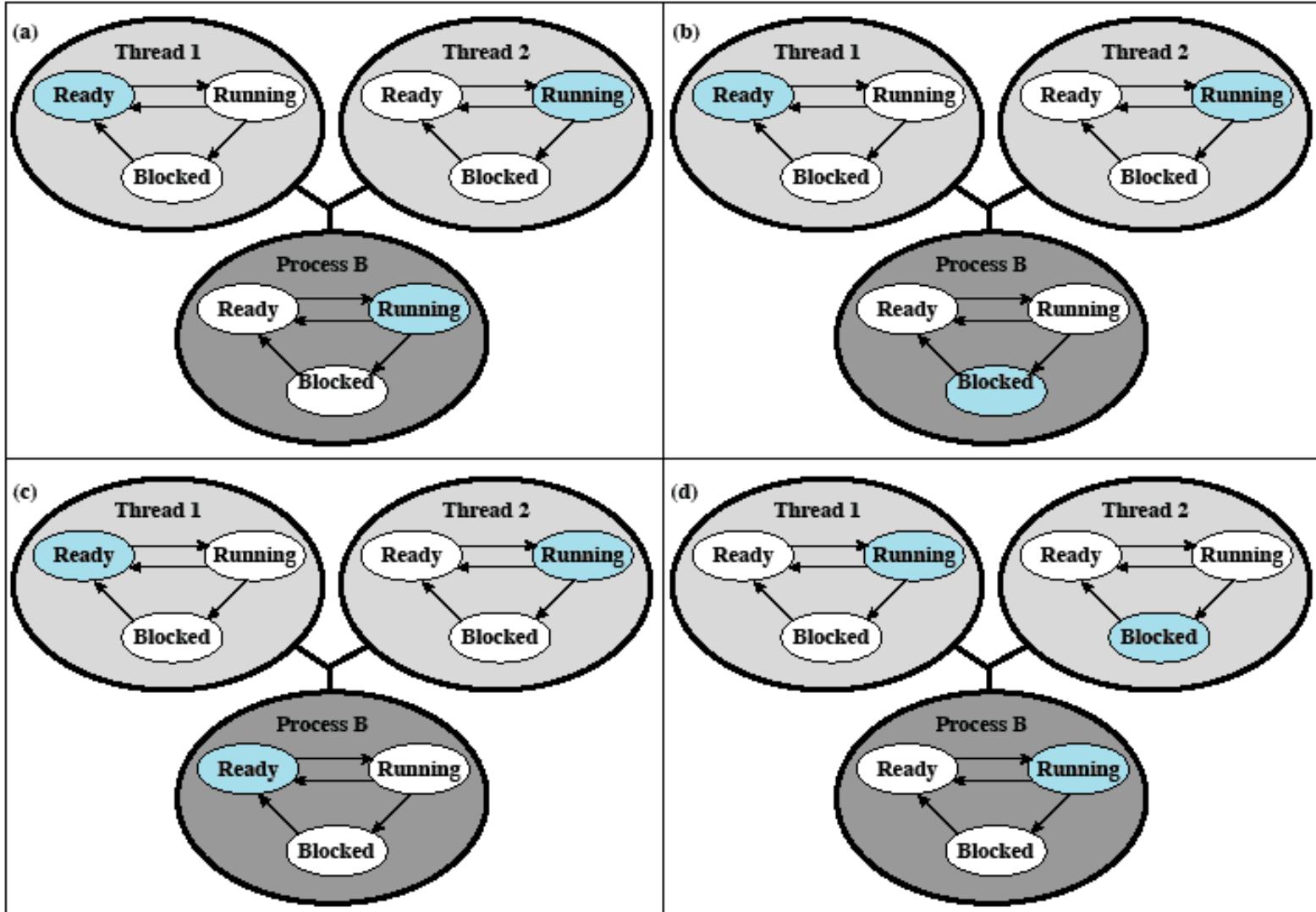
- Sav posao upravljanja nitima vrši aplikacija
- Kernel nije svestan postojanja niti
- Niti se kreiraju u okviru aplikacije korišćenjem biblioteke niti
- Biblioteka sadrži kod za
 - pravljenje i uništavanje niti
 - prosleđivanje poruka i podataka između niti
 - raspoređivanje niti
 - čuvanje i oporavljanje sadržaja niti
- Sve aktivnosti se odvijaju u korisničkom prostoru **unutar jednog procesa**



Raspoređivanje niti korisničkog nivoa

- Kernel raspoređuje procese i dodeljuje im stanje
- Niti imaju svoje stanje koje je „logička“ kategorija i nije direktno povezana sa stanjem procesa
- Stanje niti označava status niti unutar procesa
- Npr. stanje niti „Izvršavanje“
 - znači da je nit aktivna u okviru procesa
 - ne mora nužno da znači da se nit trenutno izvršava na procesoru
 - kada procesor kreće da izvršava proces, izvršavaće se nit u stanju „Izvršavanje“ iz tog procesa

Raspoređivanje niti korisničkog nivoa

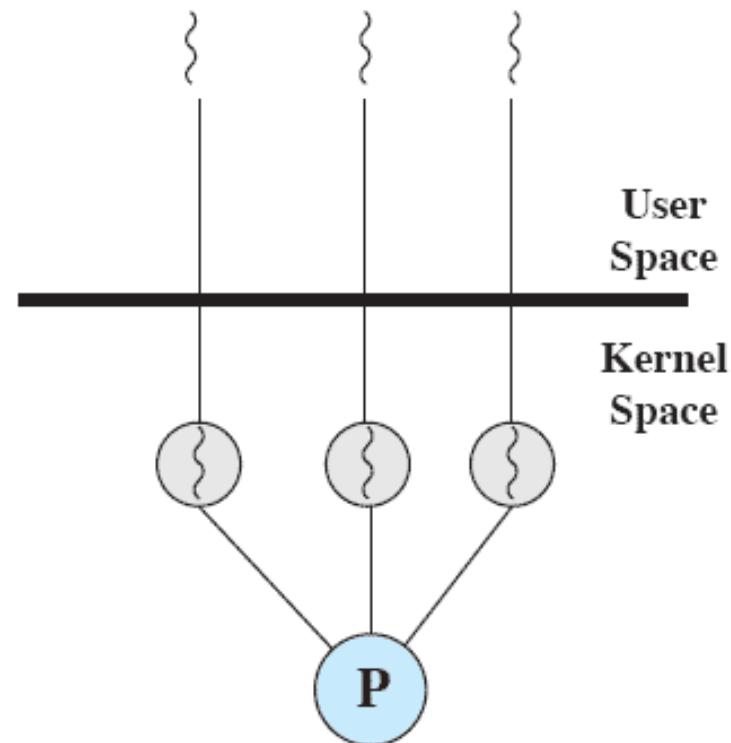


Prednosti i mane ULT

- Prednosti
 - Komutacija niti ne zahteva prelazak u režim kernela
 - Aplikacija može da implementira svoj algoritam raspoređivanja niti
 - ULT se mogu izvršavati na svakom operativnom sistemu
- Mane
 - kada se proces blokira, sve niti istog procesa postaju blokirane
 - ne može se iskoristiti multiprocesiranje, jer su sve niti jedan proces pa se izvršavaju na jednom procesoru

Niti nivoa kernela (KLT)

- Aplikacija inicira kreiranje niti
- Svo upravljanje nitima obavlja kernel
- Kernel održava podatke
 - za proces u celini
 - za svaku pojedinačnu nit
- Kernel vrši raspoređivanje na bazi niti



Prednosti i mane KLT

□ Prednosti

- Kernel može istovremeno da rasporedi više niti istog procesa na više procesora
- Ako je jedna nit u procesu blokirana, kernel može da rasporedi drugu nit istog procesa

□ Mane

- prebacivanje sa jedne na drugi nit istog procesa zahteva prelazak u režim kernela
- zato je komutacija KLT za red veličine sporija od komutacije ULT

Kombinovani pristupi

- Pravljenje niti i najveći deo upravljanja i raspoređivanja vrši se u korisničkom prostoru
- Višestruki ULT se mogu preslikati u neki (manji ili jednak) broj KLT

