Operativni sistemi: interna struktura i principi dizajna

Poglavlje 4
Niti

Sedmo izdanje

William Stallings

Procesi i niti

Tradicionalni procesi imaju dve karakteristike:

Vlasništvo nad resursima

Proces obuhvata virtuelni adresni prostor u kome se čuva slika procesa

- OS obezbeđuje zaštitu od neželjenog međusobnog ometanja procesa u pogledu resursa

Raspoređivanje/izvršavanje

Prati tok izvršavanja koji se može preplitati sa drugim procesima

- proces ima stanje izvršavanja (Izvršava se, Spreman itd.) i prioritet dodele procesora, a OS ga raspoređuje i dodeljuje mu procesor
- Tradicionalni procesi su *sekvencijalni*: *jedan* tok izvršavanja



Procesi i niti

- ***Višenitnost (multithreading)*** - sposobnost OS-a da podrži više konkurentnih tokova izvršavanja unutar jednog procesa
- Jedinica vlasništva nad resursima naziva se ***proces*** ili ***zadatak***
- Jedinica dodele procesora naziva se ***nit*** ili ***laki proces***

Jednonitni pristupi

- Jedan tok izvršavanja po procesu, pri čemu koncept niti ne postoji, naziva se jednonitni pristup
- MS-DOS i neke verzije UNIX-a podržavali su samo ovu vrstu procesa.

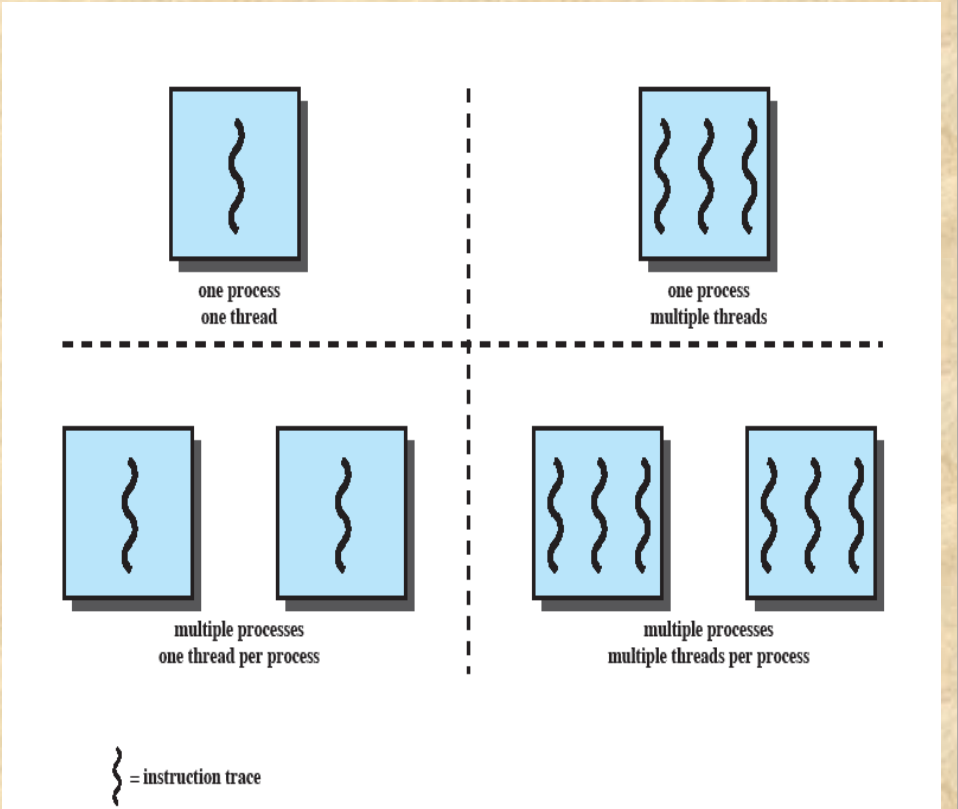


Figure 4.1 Threads and Processes [ANDE97]

Višenitni pristupi

- Desna polovina slike 4.1 prikazuje višenitne pristupe
- Java izvršno okruženje je sistem sa *jednim* procesom i više niti; Windows i neke verzije UNIX-a podržavaju *više* višenitnih procesa.

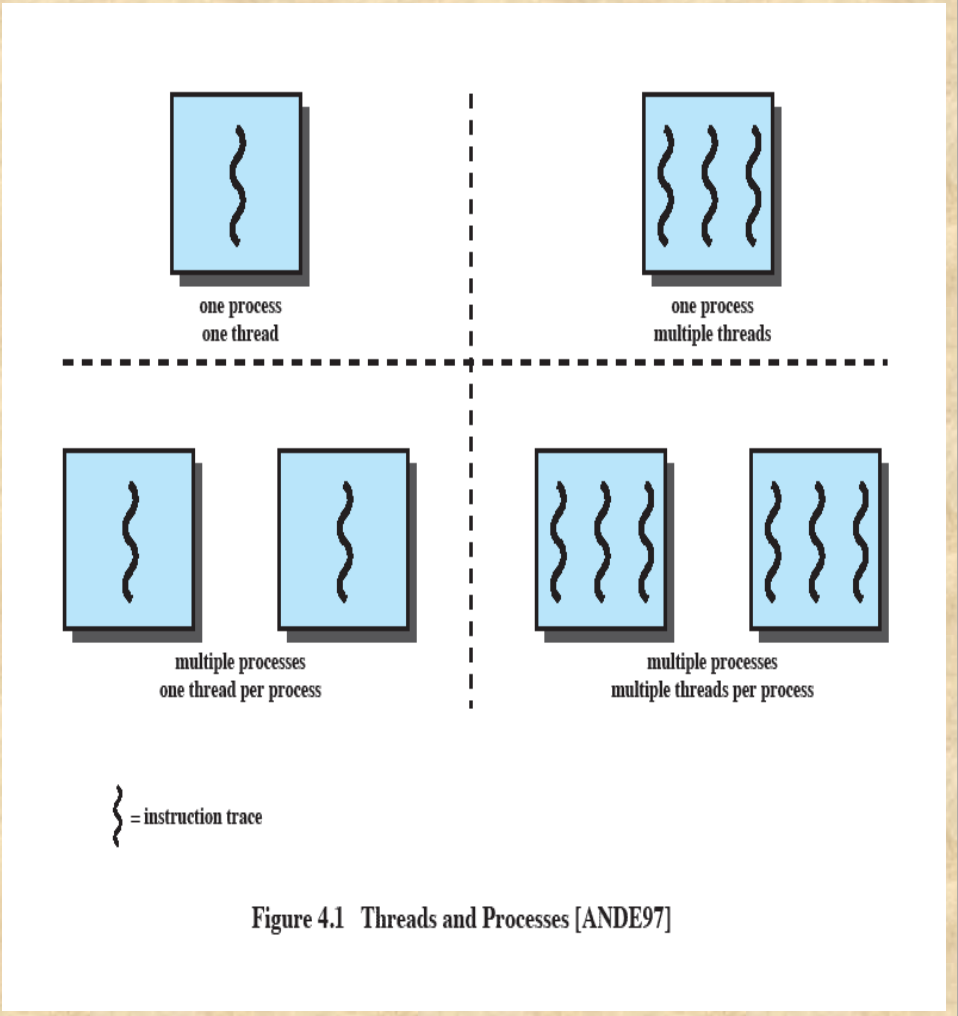


Figure 4.1 Threads and Processes [ANDE97]

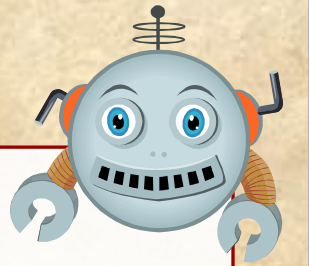
Procesi

- U višenitnom okruženju proces je jedinica koja poseduje resurse i jedinica zaštite.
  - tj. OS obezbeđuje zaštitu na nivou procesa
- Procesi imaju
  - Virtuelni adresni prostor koji sadrži sliku procesa
  - Zaštićen pristup
    - procesorima
    - drugim procesima
    - datotekama
    - U/I resursima



Jedna ili više niti u procesu

Svaka nit ima:



- stanje izvršavanja (Izvršava se, Spreman itd.)
- sačuvan kontekst niti dok se ne izvršava (TCB)
- stek izvršavanja
- statički prostor za lokalne promenljive niti
- pristup deljenoj memoriji i resursima svog procesa (sve niti procesa ih dele)

Niti naspram procesa

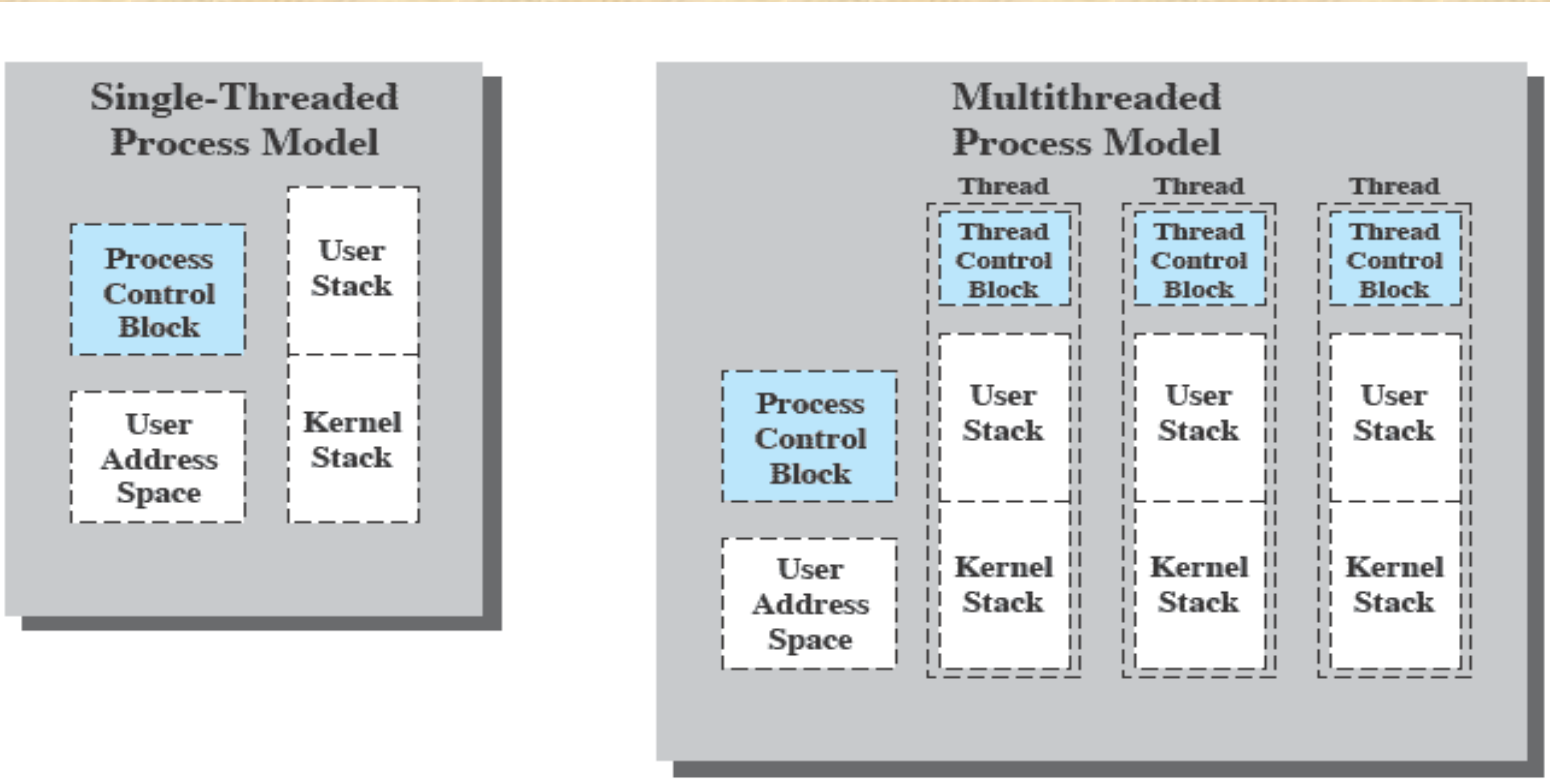
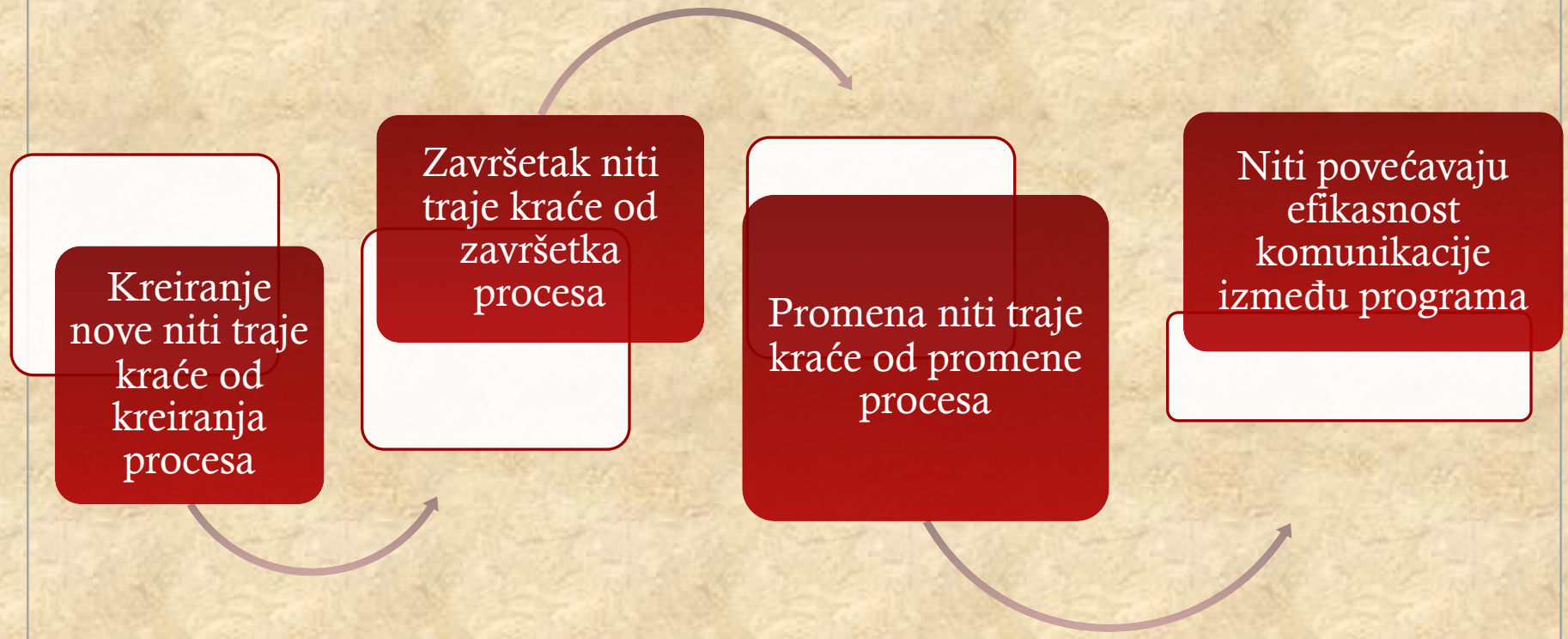


Figure 4.2 Single Threaded and Multithreaded Process Models

Prednosti niti



Upotreba niti u jednokorisničkom sistemu

- Rad u prvom planu i u pozadini
- Asinhrona obrada
- Brzina izvršavanja
- Modularnost programa



Niti

- U OS-u koji podržava niti, raspoređivanje i dodela procesora obavljaju se na nivou niti

Većina informacija o stanju koje se odnose na izvršavanje čuva se u strukturama podataka na nivou niti

- suspendovanje procesa podrazumeva suspendovanje svih njegovih niti
- završetak procesa završava sve niti



Stanja izvršavanja niti

Ključna stanja niti su:

- Izvršava se
- Spreman
- Blokiran

Operacije koje menjaju stanje niti su:

- Stvaranje
- Blokiranje
- Deblokiranje
- Završetak

Višenitnost na jednom procesoru

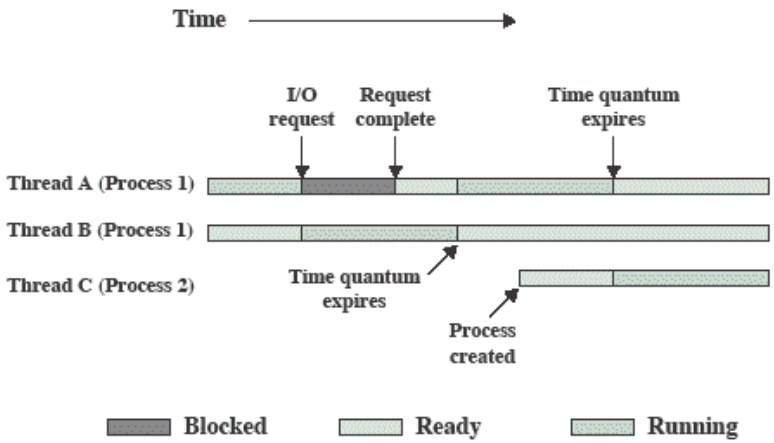
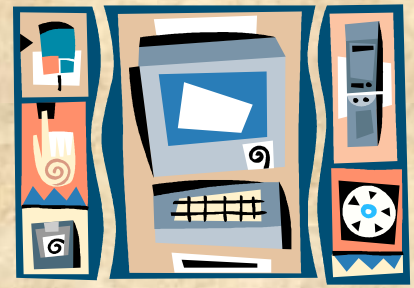


Figure 4.4 Multithreading Example on a Uniprocessor



Sinhronizacija niti

- Neophodno je sinhronizovati aktivnosti različitih niti
  - sve niti procesa dele isti adresni prostor i ostale resurse
  - svaka promena resursa koju izvrši jedna nit utiče na ostale niti istog procesa

Odnos između niti i procesa

Threads:Processes	Description	Example Systems
1:1	Each thread of execution is a unique process with its own address space and resources.	Traditional UNIX implementations
M:1	A process defines an address space and dynamic resource ownership. Multiple threads may be created and executed within that process.	Windows NT, Solaris, Linux, OS/2, OS/390, MACH
1:M	A thread may migrate from one process environment to another. This allows a thread to be easily moved among distinct systems.	Ra (Clouds), Emerald
M:N	Combines attributes of M:1 and 1:M cases.	TRIX

Tabela 4.2 Odnos između niti i procesa

Više jezgara i višenitnost

- Višenitnost i višejezgarni čipovi mogu poboljšati performanse aplikacija sa visokim stepenom paralelizma
- Primeri su igre, simulacije itd.
- Performanse ne rastu nužno linearno sa brojem jezgara …

Amdalov zakon

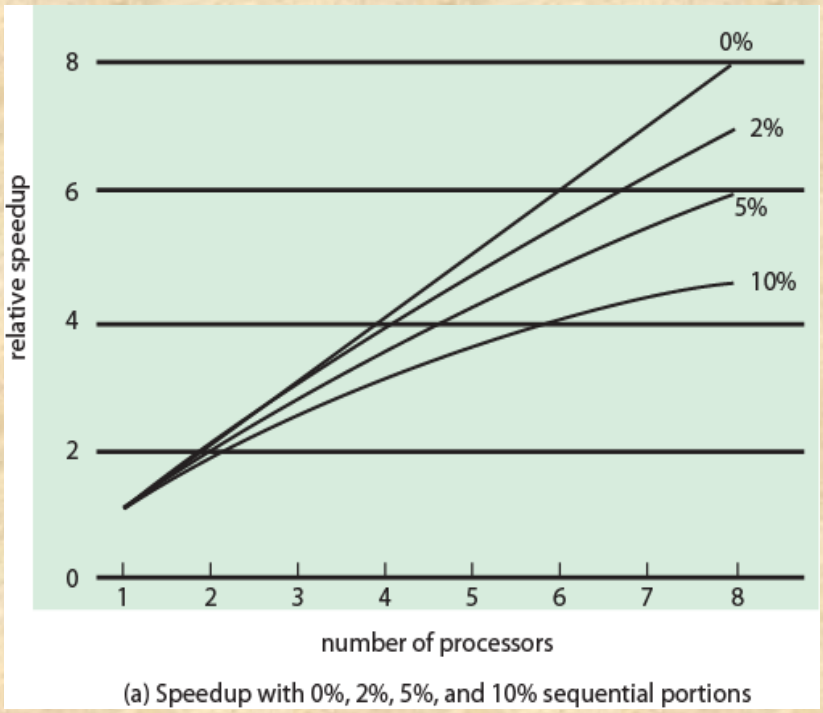
- Ubrzanje zavisi od količine koda koji se mora izvršavati sekvencijalno
- Formula:

$$\text{Ubrzanje} = \frac{\text{vreme na jednom procesoru}}{\text{vreme na N || procesora}}$$

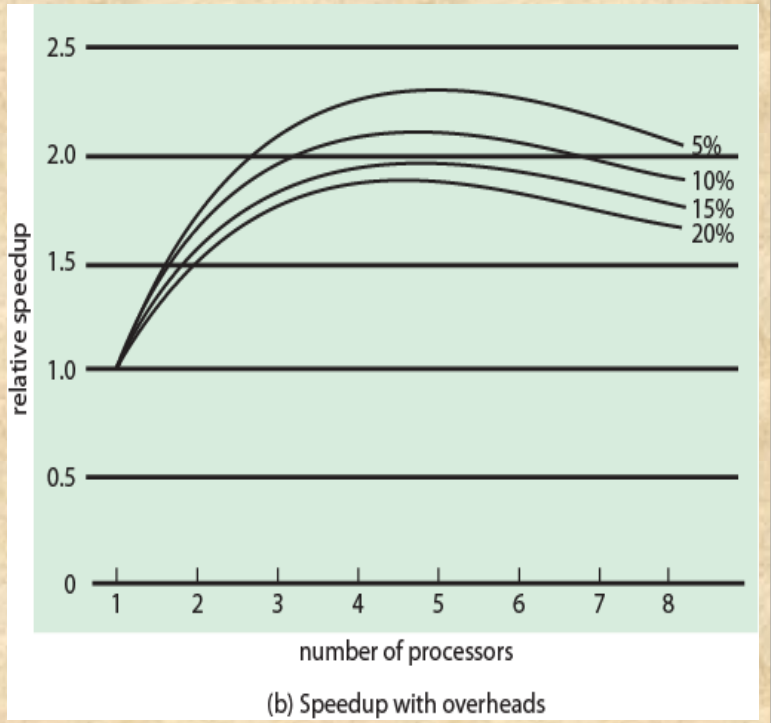
$$= \frac{1}{(1 - f) + f / N}$$

(gde je f udeo koda koji se može paralelizovati)

Uticaj više jezgara na performanse



Slika 4.7 (a)



Slika 4.7 (b)

Procesi u sistemu Windows

Procesi i servisi koje pruža jezgro sistema Windows relativno su jednostavni i opšte namene

- implementirani kao objekti
- kreiraju se kao novi proces ili kao kopija postojećeg
- izvršni proces može sadržati jednu ili više niti
- i procesi i objekti niti imaju ugrađene mogućnosti sinhronizacije



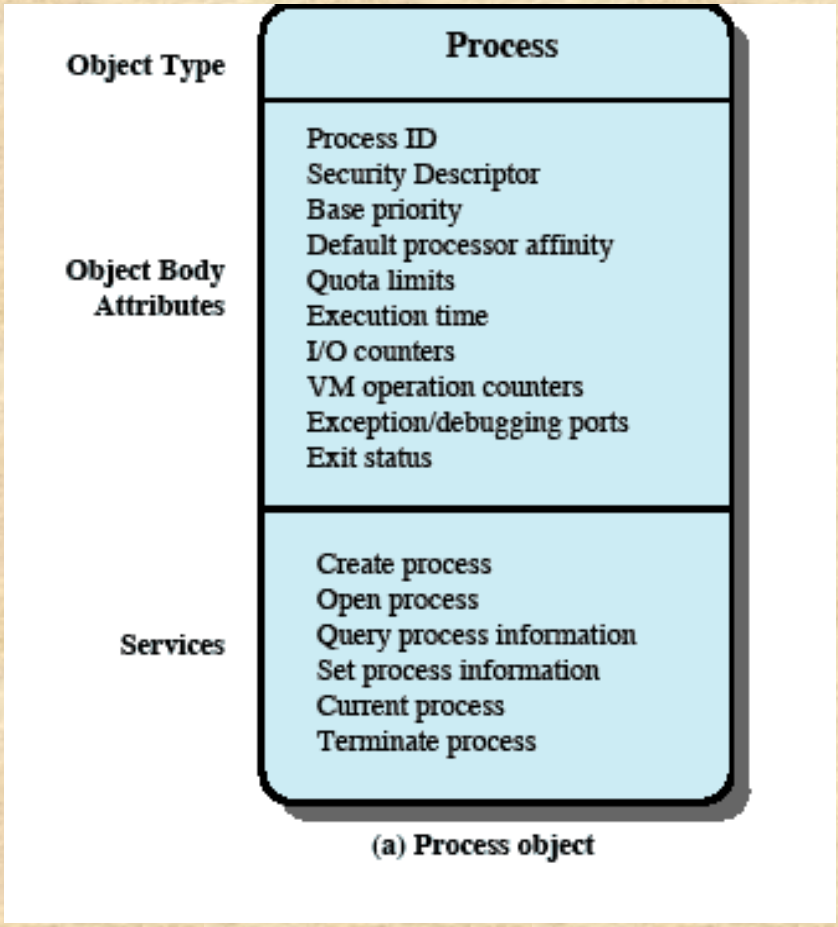
Objekti procesa i niti

Windows koristi dve vrste objekata vezanih za procese:

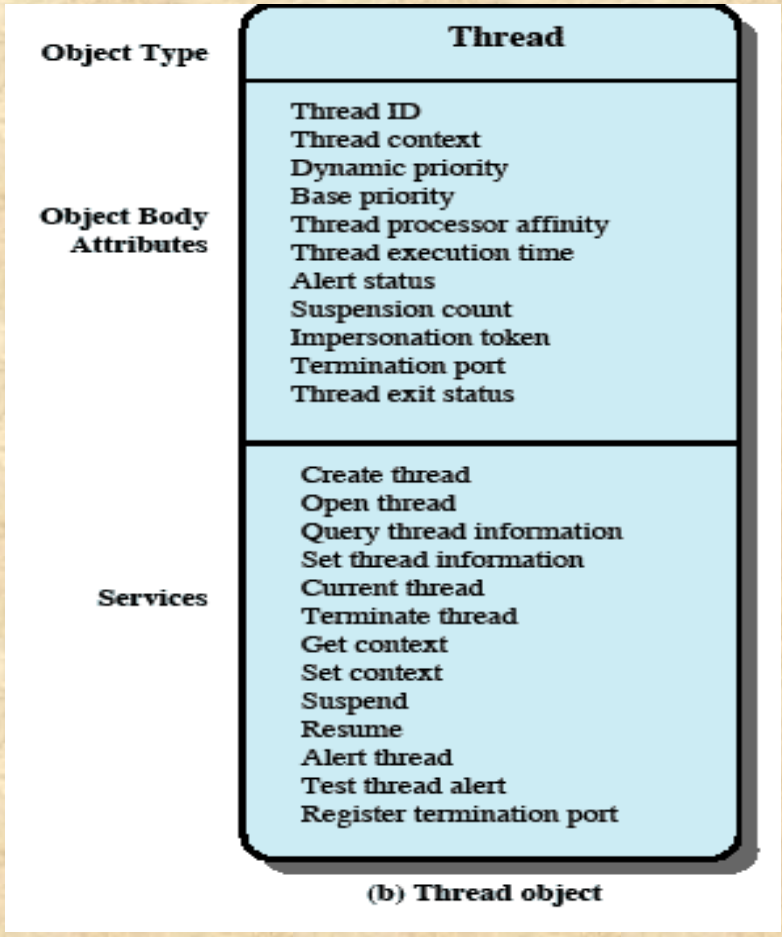
Procesi	Niti
• entitet koji odgovara poslu korisnika ili aplikaciji i poseduje resurse	• sekvencijalna jedinica posla koja dobija procesor; može biti prekinuta



Objekti procesa i niti u sistemu Windows



(a) Process object

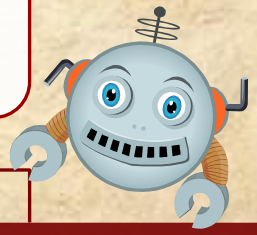


(b) Thread object

Višenitni proces



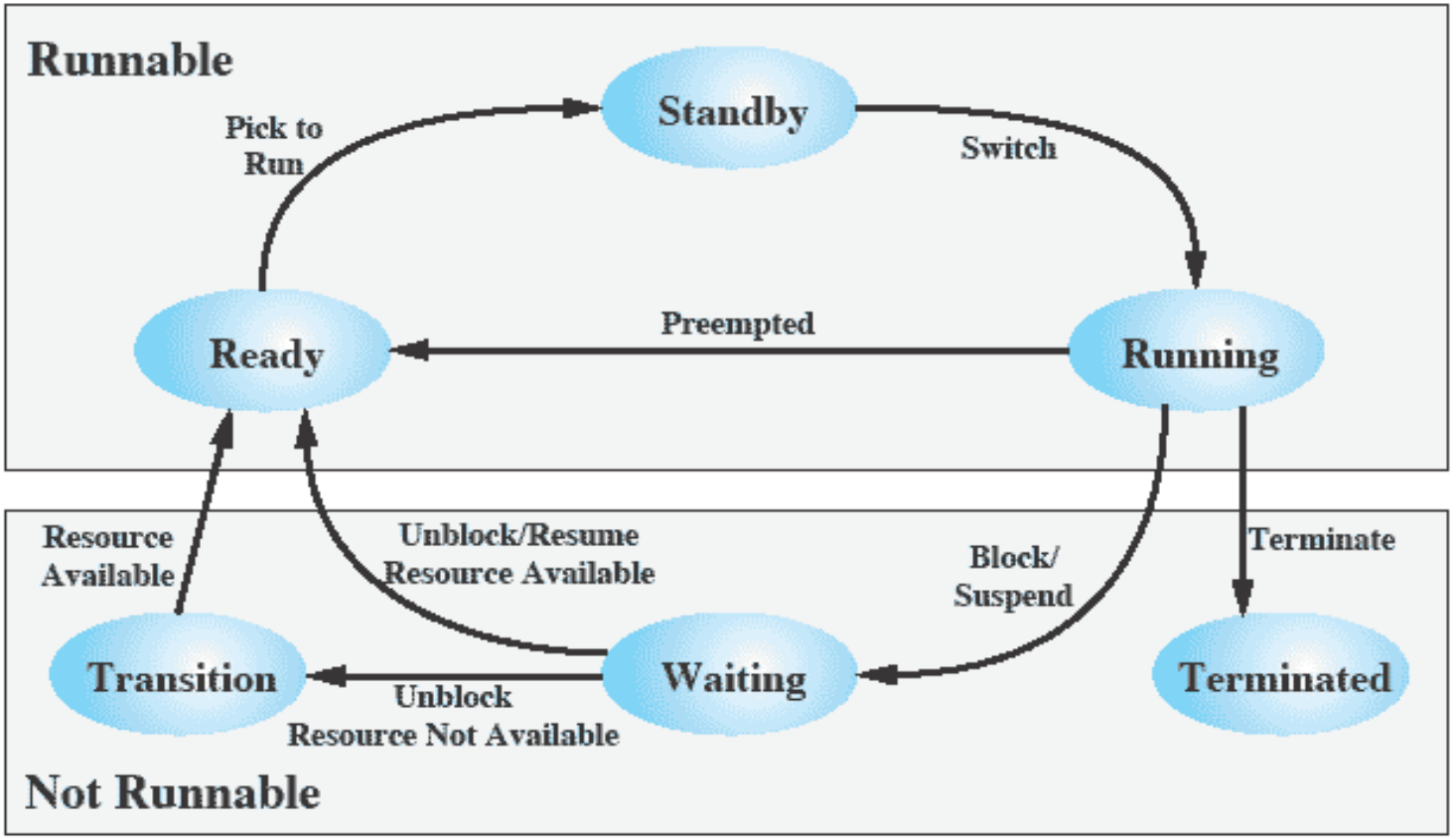
Postiže konkurentnost bez dodatnih troškova korišćenja više procesa



Niti istog procesa mogu razmenjivati informacije preko zajedničkog adresnog prostora i imaju pristup deljenim resursima procesa

Niti različitih procesa mogu razmenjivati informacije preko deljene memorije uspostavljene između ta dva procesa

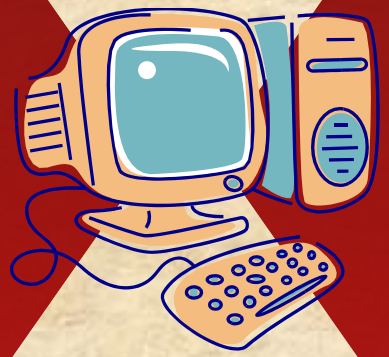
Stanja niti



Slika 4.12 Stanja niti u sistemu Windows

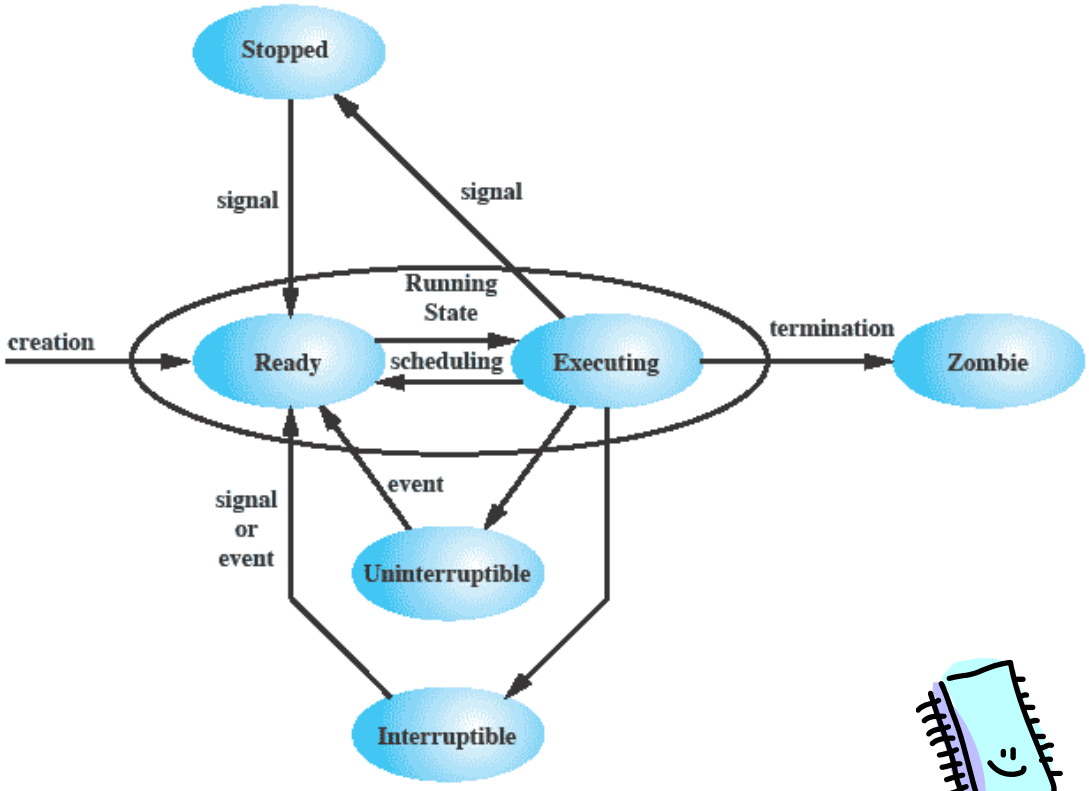
Zadaci u sistemu Linux

Proces, odnosno zadatak, u sistemu Linux predstavljen je strukturom podataka task_struct



Ova struktura sadrži informacije iz više kategorija

Model procesa/niti u sistemu Linux



Slika 4.16 Model procesa/niti u sistemu Linux

Niti u sistemu Linux

