

## Programska podrška mjernih i procesnih sustava

Branko Jeren i Predrag Pale

Fakultet elektrotehnike i računarstva  
Zavod za elektroničke sustave i obradbu signala

---

---

---

---

---

---

---

---

## Procesi

osnovni element multiprograminga  
(multi-tasking, multi-user, timesharing)

---

---

---

---

---

---

---

---

## Što je proces ?

program ----> recept, naputak  
+ podaci  
+ kontrola  
• program counter  
• stack pointer  
• register state  
+ izvođenje  
= proces ----> aktivnost, događanje

---

---

---

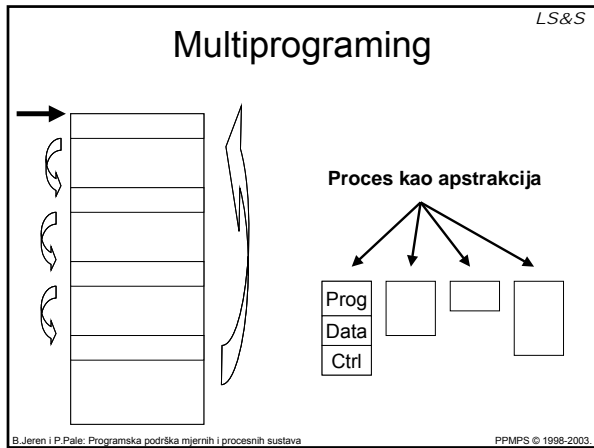
---

---

---

---

---




---

---

---

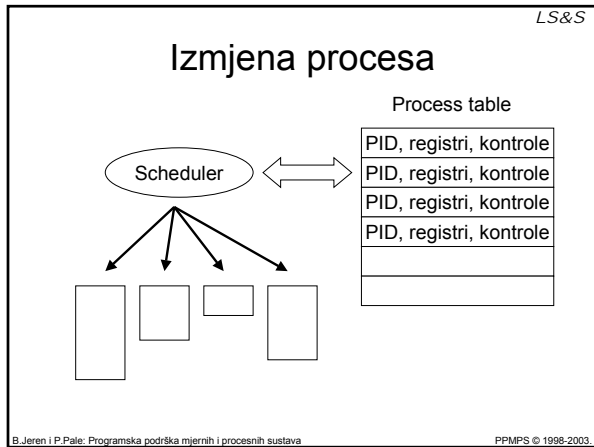
---

---

---

---

---




---

---

---

---

---

---

---

---

- ## Stanja procesa LS&S
- izvođenje (*running*)
    - procesor zaista izvodi program
  - čekanje (*blocked*)
    - ne može se nastaviti izvođenje, jer se čeka na vanjski (izvan uticaja procesa) događaj
  - spremnost (*ready*)
    - svi su (vanjski) uvjeti zadovoljeni, čeka se na slobodan procesor
- B. Jeren i P. Pale: Programaska podrška mjernih i procesnih sustava PPMPS © 1998-2003

---

---

---

---

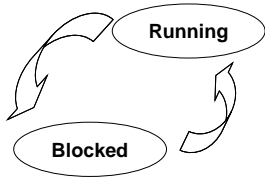
---

---

---

---

### Prijelaz među stanjima (single-tasking)




---

---

---

---

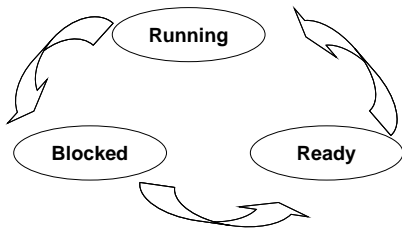
---

---

---

---

### Prijelaz među stanjima (multiprograming)




---

---

---

---

---

---

---

---

### Scheduling: zadatak

- odabrati slijedeći posao za izvođenje
- izabire između spremnih poslova
- odluka o izboru je politika, a ne tehnika
- o politici ovisi vrsta schedulera
  - zapravo scheduling algoritma

---

---

---

---

---

---

---

---

## Scheduling: zahtjevi

LS&S

- Učinkovitost (*efficiency*) [skupi CPU]
  - CPU radi koristan posao 100% vremena
- Pravednost (*fairness*) [korisnik plaća obradu]
  - svi procesi dobiju pravedan dio CPU vremena
- Vrijeme odziva (*response time*) [skupi rad korisnika]
  - interaktivni korisnik ne želi čekati odziv na svoju akciju
- Brzina izvođenja (*turnaround*) [velike obrade pod.]
  - velike obrade podataka da što brže završe
- Propusnost sustava (*throughput*) [cijena pogona]
  - što veći broj ukupno obavljenih poslova

B.Jeren i P.Pale: Programska podrška mjernih i procesnih sustava

PPMPS © 1998-2003.

---

---

---

---

---

---

---

---

## Vrste schedulera

LS&S

- Round Robin
- Priority
- Multiple Queues
- Shortest Job First
- Guaranteed
- Policy Based
- Two-level

B.Jeren i P.Pale: Programska podrška mjernih i procesnih sustava

PPMPS © 1998-2003.

---

---

---

---

---

---

---

---

## Round Robin

LS&S

- jednostavan i vrlo često korišten
- svi procesi imaju isti **quantum** vremena
- izvodi se prvi proces s popisa spremnih
  - kad upravo izvođenom procesu istekne quantum
  - kad upravo izvođeni proces ode u blokirano stanje
- problem je odrediti veličinu quantuma
  - prihvatljiv omjer prema trajanju process switch-a
  - prihvatljivo najdulje vrijeme odziva
- svi su mu procesi jednako važni

B.Jeren i P.Pale: Programska podrška mjernih i procesnih sustava

PPMPS © 1998-2003.

---

---

---

---

---

---

---

---

## Priority

LS&S

- svakom se procesu dodjeljuje **prioritet**
- među spremnim procesima, izvodi se onaj s najvećim prioritetom
- visoki prioriteti bi mogli potpuno zagušiti niže
- svako izvođenje smanjuje trenutni prioritet
- korisniku daje mogućnost definiranja prioriteta
- varijanta: klase prioriteta
  - prioritetno se odlučuje koja će klasa procesa biti na redu
  - unutar klase se koristi Round Robin

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Multiple Queues

LS&S

- ako svi procesi tipično ne stanu u memoriju
  - puno vremena troši prebacivanje iz RAM na disk i nazad
- intenzivnim procesima treba dati veliki quantum
- napravljene priority clases
  - najniža koristi 1 quant
  - slijedeća 2 quanta
  - pa 4 itd.
- ako proces iskoristi svoj cijeli quant bez blokiranja, prelazi u višu klasu

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Shortest Job First

LS&S

- za ne-interaktivne procese
- najprije se izvodi proces koji će ukupno najkraće trajati
- najbolje ukupno CPU vrijeme i prosječno vrijeme
- problem je predvidjeti koliko će proces ukupno trajati
  - predviđanja na bazi dosadašnjeg ponašanja (*aging*)

$$T = a \cdot T_0 + (1-a) \cdot T_1 \quad (\text{npr. } a=0.5)$$

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Guaranteed

- jamstvo na performansu
- najlakše je obećati resurs
  - npr. CPU vrijeme
  - nešti je teže obećati, max. vrijeme izvođenja
- potrebno je voditi evidenciju o ukupnom korištenju resursa, za svaki proces odvojeno

---

---

---

---

---

---

---

---

## Policy Based

- daje mogućnost korisniku da utječe na: redoslijed, prioritete i resurse
- potpuno odvojen od samog mehanizma schedulinga
- primijenjen standardno na UNIXu
  - ali korisnik može samo smanjivati prioritet svojim procesima
  - naredba "nice"

---

---

---

---

---

---

---

---

## Two-level

- ako procesi moraju ići na disk
- skupine procesa se izbacuju na disk
- scheduler se bavi samo onima u memoriji
- scheduler više razine odlučuje koja skupina ide s diska u RAM

---

---

---

---

---

---

---

---

## Kako odabrati scheduler

LS&S

- ovisi o
  - vrsti poslova koje obavljamo
  - tipu korisnika i njihovim potrebama
  - prioritetima vlasnika računala
- većina računala opće namjene
  - ima više vrsta poslova i korisnika
  - pa nema idealnog schedulera
- za sasvim posebne slučajeve
  - možemo napisati svoj scheduler
- novi UNIX-i
  - dolaze s više vrsta schedulera
  - koje možemo sami konfigurirati u kernel

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Niti (threads)

LS&S

- Procesi-unutar-procesa
  - Lightweight processes
- Paralelni procesi
- U istom adresnom prostoru
- Zajednički programski dio
- Ali svaka nit ima svoje:
  - Registre
  - Program counter
  - Stanje procesa
  - Stog (stack)

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Prednost niti nad procesima

LS&S

- Konkurentan pristup resursima jednog procesa.
- Više paralelnih niti izvođenja u istom adresnom prostoru
- Paralelizam izvođenja na razini procesa.
- Postizanje paralelizma izvođenja uz pomoć više procesa je nezgrapno i sporo.
- Jedan se proces izvodi brže na računalu s više procesora

B.Jeren i P.Pale: Programaska podrška mjernih i procesnih sustava

PPMPS © 1998-2003

---

---

---

---

---

---

---

---

## Razlika između procesa i niti

- Proces ima samo jednu nit
  - odvojeni adresni prostor od svih drugih procesa
- Niti dijele adresni prostor i resurse
  - npr. otvorenu datoteku
- Niti jedne drugima izravno mogu mijenati varijable, bez IPC mehanizama
- Prebacivanje procesora s niti na nit je puno brže, jer ne treba MMU

---

---

---

---

---

---

---

---

## Nedostaci i problemi niti

- Pogreške programera
- Zbog zahvata u “tuđu” memoriju i stanja resursa
  - Mijenjanje varijabli
  - Zatvaranje otvorenih datoteka

---

---

---

---

---

---

---

---

## Thread safe library

- Biblioteka korisničkih funkcija
  - koja podržava niti
- Ako jedna nit pozove funkciju
  - čije se izvršavanje ne dovrši
- Kad u izvođenju krene druga nit
  - koja pozove istu funkciju
- Statičke varijable funkcije
  - ne smiju biti promijenjene

---

---

---

---

---

---

---

---



## Razina na kojoj se podržavaju niti

- Aplikacijska
  - Funkcionalnost koju podržavaju neki moduli u korisničkoj biblioteci
  - Operacijski sustav to ne zna
  - Ako jedna nit ode na spavanje, odlazi cijeli proces i sve ostale aktivne niti
  - Sporije se izvršavaju one niti koje su u procesima s više niti
- Jezgrena
  - Sporiji context switching
  - Učinkovitija raspodjela resursa na niti
- Hibridna
  - M aplikacijskih niti je podržano od N jezgrenih
  - $M > N$
  - Dva schedulera: za procese i za niti
  - Međusobno komuniciraju

---

---

---

---

---

---

---

---

---

---

## Kuda dalje ?

- paralelni procesi imaju potrebu za međusobnom komunikacijom
  - => IPC = InterProcess Communication
- paralelni procesi često koriste neki zajednički resurs
  - može doći do "race condition"
  - ili do "deadlocka"

---

---

---

---

---

---

---

---

---

---

## Programska podrška mjernih i procesnih sustava

www.zesoi.fer.hr  
(<http://www.ZESOI.FER.hr/hrzesoi/dodip/ppmips.htm>)

ppmps@zesoi.fer.hr

---

---

---

---

---

---

---

---

---

---