

OPERACIJSKI SUSTAVI

OPCI SEMAFOR

- Dijkistra - OS
- današnji - Osem

$$- Os[i].v = 1 - NP[i]$$

- sve dretve pros proglašuju pripravnina

- u svim kodova se stave sinkronizacijske funkcije

- vrijednost Osem IN

- za početku vrijednost M , ispitat će se vrijednost na dretvi: tek će se $M+1$ blokirati

KAŠNJENJE

- dretva kojoj se definira kašnjivanje, treba biti prvo u listi Aktivna_D da bi joj se period mogao definirati
- dretva može zatražiti odgazanje neke druge dretve za M perioda (ta dretva MORA biti u listi Pasiva_D)
- hardverski prekid (otkucaj sata) oslobođava dretvu iz liste Odgodene_D (za bivanju semafor taj prekid je softverski)

ULAZNO - IZLAZNE FUNKCIJE

- signaliziraju prekide = hardverski

- struktura podataka jezgre NORAJU biti u dijeljenom spremniku
- isključivanje u višeprocesorskom sustavu = TAS
- OBRADA_JEZGRE = svaka jezgreva funkcija ju mora ispunjavati (samo se 1 jezgreva funkcija može nat. izrišavati)
- svaki procesor ima svoju listu Aktivna_D
- dretve nisu različiti procesori (bez obzira na komogem sustav)
- mogu utjecati međusobno jedna na drugu: na nijenos izvodjenje
- funkcija ispitati sve koristi varijable i broj dretve: broj procesora
- zastavica TAS kada pročita 0, u sljedećem ciklusu upše 1
- u funkciji Postaviti_Bsem, dretva koja čeka na bsem vraća se u listu Pripravnina_D stoga je procesora (index procesora ne prenosi se kao parametar jer ne znamo ima li na tojuci procesoru dretva koja čeka na bsem, mogu se broj procesora prošta u opisujući dretve)

- SEMAFORI = jezgreve mehanizmi za sinkronizaciju i međusobno isključivanje dretvi; sastoje se od varijable i reda opisujuća dretvi koje nisu uspjeli prći
- Bsem[i].v = 1 \Rightarrow semafor prolazan
- Bsem[i].v = 0 \Rightarrow semafor neprolazan
- BSEM ima 3 stanja \Rightarrow PROLAZAN, NEPROLAZAN + PRAZAN RED, NEPRAZAN + RED

tel +385 1 3667 273, 3667 275

fax +385 1 3667 287, 3667 622

info@koncar-mes.hr

- $Ose[i].v \in \mathbb{Z}$
- > 0 ... semafor prolazan
- ≤ 0 ... \neg neprolazan
- $Osem[i].v \in \mathbb{N}_0$
- > 0 ... semafor prolazan
- $= 0$... \neg neprolazan

PROLAZAN

NEROLAZAN, RED PRAZAN

NEROLAZAN, RED NEPRAZAN

Cekaj_Bsem

$v = 0;$
prodi semafor;

prijedti u red Bsem;

Postavi_Bsem

$(v = 1)$

$v = 1;$

preustvari prvi opisnik reda
Bsem u Pripravne_D;

Cekaj_Os

$v--;$
prodi semafor;

$v--;$
prijedti u red Os;

Postavi_Os

$v++;$

$v++;$

$v++;$
ako ($v == 0$) preustvari prvi
opisnik u red Pripravne_D;

Cekaj_Osem

$v--;$
prodi semafor;

prijedti u red Osem;

Postavi_Osem

$v++;$

$v++;$

preustvari prvi opisnik iz
reda Osem u Pripravne_D;

Z1 Dretva Dx

dok je (1)

čekaj - Bsem (s); a

piši ("Px");

postavi - Bsem (s), b

piši ("zx");

dretva se može pretvoriti ovdje i samo ovdje

U jednoum procesorskom računalu pokreće se sustav dretvi D1, D2, D3, D4 s prioritetima jednaku indexu (max. 4).

Drete obavljaju kod odika Dx

Red pripravnih dretvi i red semafora su prioritetski redovi.

Aktivna je dretva ona koja je prva u redu pripravnih.

Prije pokretanja sustava dretvi semafor s je bio nepostavljen. Nakon nekog vremena kde dretve se uđu u red semafora s. Ako se tada pozove procedura Postavi - Bsem (s), što će se ispisati na zaslonu?

RJEŠENJE:

Bsem.v	RED SEMAFORA	RED PRIPRAVNIH	ISPIS P	ISPIS Z
Ø	4 3 2 1	-		
Ø	3 2 1	4a	P4	Postavi - Bsem
Ø	2 1	4b 3a		Postavi - Bsem D4
Ø	4 2 1	3a	P3	Cekaj - Bsem D4
Ø	2 1	4a 3b	P4	Postavi - Bsem D3
Ø	1	4b 3b 2a		Postavi - Bsem D4
Ø	4 1	3b 2a		Cekaj - Bsem D4
Ø	4 3 1	2a	P2	Postavi - Bsem D3
Ø	3 1	4a 2b	P4	Cekaj - Bsem D4
Ø	1	4b 3a 2b		Postavi - Bsem D4
Ø	4	3a 2b		Cekaj - Bsem D4
Ø	1	4a 3b 2b	P4	Postavi - Bsem D3
Ø	-	4b 3b 2b 1a		Postavi - Bsem D4
Ø	4	3b 2b 1a		Cekaj - Bsem D4
Ø	4 3	2b 1a		Postavi - Bsem D2
Ø	4 3 2	1a	P1	Postavi - Bsem D1
Ø	3 2	4a	P4	Postavi - Bsem D4
Ø	2	4b 3a	1b	Cekaj - Bsem D4
Ø	4 2	3a	1b	Postavi - Bsem D3
Ø	2	4a 3b	1b	Postavi - Bsem D4
Ø	-	4b 3b 2a	1b	Cekaj - Bsem D4
Ø	4	3b 2a	1b	Cekaj - Bsem D3
Ø	4 3	2a	P2	Postavi - Bsem D2
Ø	3	4a	2b	Postavi - Bsem D4
Ø	-	4b	3a 2b	Cekaj - Bsem D4
Ø	4	3a	2b	Postavi - Bsem D3
Ø	-	4a	3b 2b	Postavi - Bsem D4
1		4b	3b 2b 1b	Cekaj - Bsem D4
Ø		4a	3b 2b 1b	Postavi - Bsem D4
	...			

OPERACIJSKI SUSTAVI

6. MEDUDRETVENA KOMUNIKACIJA I KONCEPCIJA MONITORA

29.10.2008.

- PROIZVODAČ = ciklička držva u kojoj se generiraju podaci, uključujući u poruke i šablonu potrošaču

- POTROŠAČ =

- zatvoreni elektronizacijski krog čine proizvođač koji proizvodi određenu poruku i potrošač koji prima upravo tu poruku

- MEDIUSPREMNIK (BUFFER) - zbog proizvoljnih brzina

- u adresnom prostoru procesora
- Neognani kapacitet - neobičan slučaj
- 2 pointer = UL i LZ
- koristi ponku ponuđu Bsem
- realno = ograničeni spremnik
- dodatni Bsem koji koristi prazne pretrice
- još jedan Bsem za zaštitu K.O.

6.2. POTPUNI ZASTOJ

- j-funkcije oduzimaju mnogo vremena

- nepoštivanu upotrebu može doći do potpunog zastoja (npr. zamjenjujuće redoslijeda pozivawaja semafora)

- UVJETI: 1. Nešto sredstvo u istom trenutku može upotrijevavati samo 1 drživa (nedostatno iskorišćivo)

2. držvi se sredstvo ne može oduzeti - ona ga otpušta samo kad ga više ne potrebuje

3. drživa drži dodjeljeno joj sredstvo dok žeka na dodjelu dodatnog sredstva

- potpuni zastoj se isklučuje eliminiranjem bilo kojeg od ta 3 uvjeta (uvjedite 3.)

- MONITORSKE FUNKCIJE = piše konzoli; za sintetizaciju, rješavanje problema

- dok je jedna držva u monitoru, druga ne može ući
- USER MODE

- držva koja vrši ne funkcije ne može biti prekinuta, druga ne može ući za to vrijeme prekida, dok prekinuta ne izadi

PRIMJER: 5 FILOZOPA

dok je (1)

{

misli;

<monitor>

uci u monitor; "K.O.

dok (neima 2 vrste) "varijable

detaj (u redu monitora); "j-fja

vrni vrste; "varijable

izadi iz monitora; "j-fja

jesti;

<monitor>

uci u monitor;

susti vrste; "varijable

javljevi: dešnovi susjed;

"ostabidi iz reda vrsta (j-fja)

izadi iz monitora;

3.11.2008.

OPERACIJSKI SUSTAVI

6. MEDUDRETVENA KOMUNIKACIJA I KONCEPCIJA MONITORA

6.X PROBLEMI SA SEMAFORIMA

PROBLEMI

- svaki semafor ispituje samo 1 uvjet
- ispitivanje semafora povezano je sa zaustavljem sredstva kojeg semafor štiti
(nije moguće prethodno davanje više ispitivanja, a onda sve zauzeti)
- opreznost od potpunog zastoja

RJEŠENJE

- MONITOR - sastoji se od : 1. strukture podataka (m. semafor, redovi, uvjeta)
 - 2. monitorских funkcija (izlode se međusobno istjivaju)
- monitorstva funkcija se izvodi u korisničkom načinu rada \Rightarrow rješava se konkretni sinkronizacijski problem, piše ju korisnik \Rightarrow zato mora biti u korisničkom načinu rada
- UNUTAR MONITORSKE FUNKCIJE - dretva ispituje neki složeni uvjet
 - ako je uvjet zadovoljen, dretva zauzima sredstva ('obavlja posao') : dozvoljava drugoj dretvi ulazak u monitor
 - ako uvjet nije zadovoljen, dretva se blokira na redu dočekog ujeta i omogućuje drugim dretvama ulazak u monitor

NEKA PRAVILA ZA MONITORSKE FUNKCIJE

- na početku se čeka, a na kraju funkcije postavlja monitorski semafor
- funkcije vrši u red : oslobodi iz reda pozvano se samo unutar monitorstkih funkcija
- uvjet se obično ispituje sa : dok je (while), a ne ako je (if)

- pregrane funkcije za ostvarivanje monitora
- monitorstke funkcije (m-funkcije)

PRIPREMA ZA 3. LABORATORIJSKU VJEŽBU

SEMAFORI

- 2 proizvodača + 1 potrošač

GLAVNI PROCES : - dodatak semafore (20sem + 1 Bsem)

- inicijalizacija semafora ($S_0, R_0 = 1$)
- zauzeti zajedničku memoriju
- inicijalizirati zajedničke varijable
- postavi funkciju obrade prekida SIGINT - osloboduje memorije (npr. ako se konač program)
- stvori procese (2 proizvodača + 1 potrošač)
- čekaj procese
- osloboditi sredstva (semafori, memorija)

SEMAFORI : o Piši - Bsem za zaštitu pisanja; proizvodači ; Piši - 1;

o PUN - Osem ; broj punih pretinaca ; PUN=5;

o PRAZAN - Osem ; broj praznih pretinaca ; PRAZAN = 0

- POMOĆNE FUNKCIJE : o uputa

o semop (<ID>, -1) ... čekaj
semop (<ID>, +1) ... postavi

- početna synchronizacija potrošača \Rightarrow učitavanje teksta zaštiti Bsem;

ZAJEDNIČKA MEMORIJA : o 2 integer varijable za pointere

o polje M[5] za međuspremnik (char)

- PINUS : min 128 bajtova ; 100 / 128 integera

PRIMJER : 1 segment za 2 miza i 2 varijable

```
double *poljeD, *D;  
integer *poljeI, *I;
```

```
int ID = službeni(...); // programi?
```

```
poljeD = (double *) službeni (ID, ...);
```

```
poljeI = (integer *) (poljeD + <veličina_polja>);
```

```
D = (double *) (poljeI + <veličina_varijable>);
```

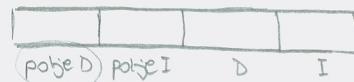
```
I = (integer *) (D + <veličina_varijable>);
```

...

```
službeni ((char *) poljeD);
```

" pointer na početak

```
službeni (ID, ...);
```



OSLOBADANJE SEMAFORA : o semafore () ;

- napisati funkciju za oslobadanje memorije i sredstva općenito prikazu korisničkog signala SIGINT
- za nepreduvoden završetak programa, koristi naredbe (upise) : IPCS, IPCRM
- proizvodaci su zaustavili makrom drugog backslasha

ipcs

ipcrm -m <broj>

ipcrm -s <broj>

MONITORI

VARIJABLE NEKUSOBNOG ISKLJUČIVANJA - djeluje kao ţsem

PRIMJER: (MUTEX)

pthread_mutex_t ključ [n]; ... identifikator
pthread_mutex_init (&ključ[i], NULL); ... inicijalizacija

udi u monitor

pthread_mutex_lock (&ključ[i]); ... ţekaj ţsem (i)

izadi iz monitora

pthread_mutex_unlock (&ključ[i]); ... postavi ţsem (i)

- ključevi moraju biti globalne varijable - da ih vide sve dretve

UVJETNE VARIJABLE = oznaka (ID) određenog uvjeta

- globalne

- ukoliko uvjet nije zadovoljen, dretva ulazi u red uvjeta te istovremeno oblikuje ključ

cond_wait (<uvjet>, <ključ>);

- druga dretva može pozvati (spurje) uvjeta :

pthread_cond_signal (<uvjet>) ... oslobođa 1 dretvu

pthread_cond_broadcast (<uvjet>) ... oslobođa sve dretve

- ako se pozove cond_signal (cond_broadcast), a u redu uvjeta nema nikoga, to će značiti sljedeći putanj za iduću dretvu koja pozove cond_wait

OSTVARENJE MONITORA U LAB. OKRUŽENJU

- ekvivalentno pozivanje funkcija

tel +385 1 3667 273, 3667 275

fax +385 1 3667 287, 3667 622

info@koncar-mes.hr

- o mutex_lock \Rightarrow uči u monitor
- o mutex_unlock \Rightarrow izadi iz monitora
- o cond_wait \Rightarrow vršiti u red uvjeta + uči u monitor
- o cond_signal \Rightarrow osloboditi iz reda uvjeta + uči u monitor

i-fja cond_signal

pdjekuti kontekst u opisnik Active_D,
ako je (red uvjeta (Luvjet)) neprazan
 | prvi opisnik iz reda \rightarrow Pripravne_D;
 osloboditi kontekst Active_D,
 omogućiti prelaz;
 i fati se iz prethodnog reda rada;

- po pozivu cond_signal, direvka koja se deblokira (ne boja posava) prenosi se u red čekanja na ulaz u monitor
- direvka koja je pozvala cond_signal ostaje u monitoru (i dalje drži mutex zatvoren; zauret semafor)

sveka vilica ima svoju varijablu / svakko vidi z vilice
U PRIPREMI?

tip podatka ptured_cond_t (uvjet)

5.11.2008.

OPERACIJSKI SUSTAVI

Zadatak: PROBLEM STAROG MOSTA

Na starom mostu smije istovremeno biti max. 3 automobila i moraju voziti u istome smjeru.

Simulirati automobile drevom AUTO i napisati multitočke funkcije:
popeti_se_na_most;
sidi_s_mosta;

AUTO (snijfer) // snijfer o/1

popeti_se_na_most (snijfer);
projedi_most;
sidi_s_mosta (snijfer);

VARIABLE !

- auto ... broj automobila na mostu
- snijfer_na_mostu
- M ... monitor, red uvjeta 1

(auto = 0)

(snijfer_na_mostu = 0)

M-funkcija popeti_se_na_most (snijfer)

```

mutex_lock(M);
while ((auto == 3) || (auto > 0) && (snijfer != snijfer_na_mostu))
    cond_wait(M, 1); // 1 red uvjeta
auto++;
snijfer_na_mostu = snijfer;
mutex_unlock(M);
  
```

M-funkcija sidi_s_mosta (snijfer)

```

mutex_lock(M);
auto--;
cond_broadcast(1); // kodi sre direkle koje čekaju
mutex_unlock(M);
  
```

=> MODEL MONITORA S PREDANJEM

- monitor M
- 2 reda uvjeta : 0, 1 // snijfer

M-funkcija kreui_na_most (snijfer)

```

uci_u_monitor(M);
dok (...){
    uvrsti_u_red(M, snijfer);
    uci_u_monitor;
    auto++;
    snijfer_na_mostu = snijfer;
}
  
```

izaci je monitora;

tel +385 1 3667 273, 3667 275

fax +385 1 3667 287, 3667 622

info@koncar-mes.hr

M-funkcija sidi_s_mesta (auger)

```

    uci_u_monitor (M);
    auti --;
    ako (auti > 0)
        | oslobodi_iz_reda (auger);
        inace
            | oslobodi_iz_reda (M, 1, auger)
            uci_u_monitor (M);
            izaci_iz_monitora (M);
    }
}
```

} 3x ??

• KOMENTAR •

- učinkovitije rješenje bri vodilo računa o tome koliko automobila čeka u određenom vijetu i posivalo je grane funkcije samo kada je to potrebno
- moguća je poziva izgradnjuvača

Zadatak : PING + PONG DRENE

U sustavu postoje 2 vrste dretvi: ping dretve koje u beskonačnoj poteci ispisuju nječ PING ; pong dretve koje ispisuju PONG.

Dretve treba synkronizirati tako da naizmjenice ispisuju PING + PONG.

- a) uporabom SEMAFORA
- b) uporabom MONITORA

SEMAFORI

PING ()
 | cekaj_lssem [0];
 | ispis;
 | postavi_lssem [1];

PONG ()
 | cekaj_lssem [1];
 | ispis;
 | postavi_lssem [0];

• POČETNE VRIJEDNOSTI =
 lssem [0]. v = 1
 lssem [1]. v = 0

MONITORI

- monitor M
- 2 reda uvjeta (0/1)

- stanje {0,1}
- vrsta {0,1} ili

a) 1 MONITORSKA FUNKCIJA

- monitor M-funkcije konstintno "sredstvo" → u ovom slučaju ispis

DRETVA ()
 | ispis (vrsta);

M-funkcija ispis (vrsta)
 | mutex_lock;
 | dok (stanje != vrsta)
 | | cond_wait (M, vrsta);
 | | ispis (...);
 | | stanje = 1 - stanje;
 | | cond_signal (1 - vrsta);
 | | mutex_unlock (M);

b) 2 MONITORSKE FUNKCIJE

DRETNA ()

PRIJE ()

ispis;

POSLJE ()

H-funkcija PRIJE (vrsta)

mutex_lock (M);

dok (stavje != vrsta)

coud_wait (M, vrsta);

stavje = -1;

mutex_unlock (M);

// zbrajam vrijednost!

H-funkcija POSLJE (vrsta).

mutex_lock (M);

stavje = 1 - vrsta;

coud_signal (1 - vrsta);

mutex_unlock (M);

// ovoj dogodi

Zadatak : Microsoft i Linux programer

U istoj zgradi radi Microsoft i Linux programer. Zgrada ima 1 restoran kojega programeri moraju dijeliti. U istom trenutku u zgradi mogu biti samo 1 vrsta programera (ili je prazan).

Štaki parametar ima sljedeći oblik:

PROGRAMER (vrsta)

↓

UDI (vrsta);

jedi;

IZADI (vrsta);

- sinkronizirati programere monitorom

- zadatok resiti bez izgradnjivanja tako da programer jedne vrste ne mogu ući ako ih je već ušlo N te vrste, a programer druge vrste

vrsta = 0 / 1

brojac

Zadatak

U istoj zgradi žive Microsoftovi programer i Linux programer. Zgrada ima 1 restoran kojega programer moraju dijeliti. U istom trenutku u restoranu smije biti samo 1 vrsta programera (ili je prazan). Svaki programer ima sljedeći oblik:

programer (vrsta)
{

udi (vrsta);

jedi;

izudi (vrsta);

}

- Synchronizirati programere monitorom;

- Zadatak rješiti bez pojave izglađnjuvaja tako da programer jedne vrste ne mogu ući u restoran ako ih je već ušlo N iste vrste, a programeri druge vrste neka

7. ANALIZA VREMENSKIH SLOZINA RAČUNALNOG SUSTAVA

7.1. UNODNA RAZMATRANJA

- DRETVA = osnovna jedinica aktivnosti unutar računalnog sustava
 - traje nekoliko godina je nepredvidivo

- OCJENA PONASANJA : 1. MATEMATICKI MODEL

2. SIMULACIJSKI MODEL

3. MERENJE PONASANJA STVARNOG SUSTAVA

- DETERMINISTICKI SUSTAV = može se predviđati

↳ periodički poslovi

- najjednostavniji → znai se kad dolazi, u kojem intervalu ; koliko traje

- P ... OPTEREDJENJE POSLUŽITELJA iskoristivost sustava

- α ... BROJ DOLAZAKA NOVIH POSLOVA U JEDINICI VREMENA

- β ... SNAGA POSLUŽITELJA - broj poslova u jedinici vremena izlazaka

$$P = \frac{\alpha}{\beta} \leq 1$$

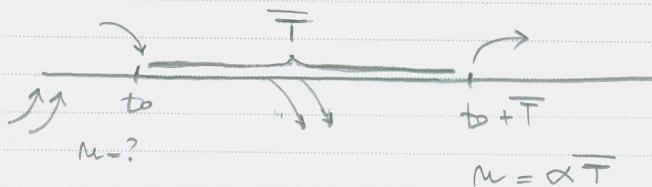
$\bar{n} = \alpha * T$ - prosječno vrijeme zadržavanja u sustavu

↳ prosječan broj

OPERACIJSKI SUSTAVI

- DOKAZ: Littleovo pravilo $\bar{m} = \alpha \bar{T}$

- primjer

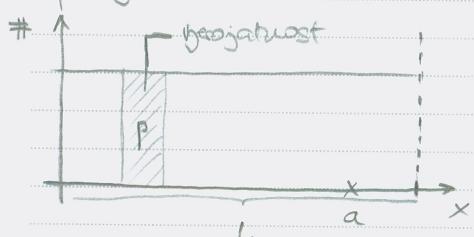


- broj poslova u sustavu u vremenu izlasta tom poslu je sustava jeduak je broju poslova koji su ušli u sustav u vremenu \bar{T} koliko se tog posla zadržao u sustavu \Rightarrow vrijedi samo za DETERMINISTIČKI MODE

RASPODJELE

- RASPODJEZA opisuje pojavljivanje slučajne varijable

- primjer: kiša



za dovoljno dugo provatraanje, na svako mjesto post će jednak broj kapi

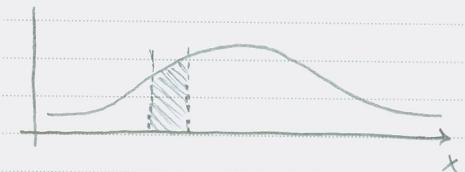
\Rightarrow UNIFORMNA (JEDNOLIKA) RASPODJEZA

- raspodjela je definirana funkcijom gustoće vjerojatnosti

- vjerojatnost da će pasti točko u $a = 0$ je x brojne vrijednosti
- zato provjeravamo vjerojatnost uutar intervala

- vjerojatnost se dobiva kao površina ispod pulcije gustoće vjerojatnosti

- primjer: stepenica



\Rightarrow GAUSSOVA (NORMALNA) RAZDIJELA

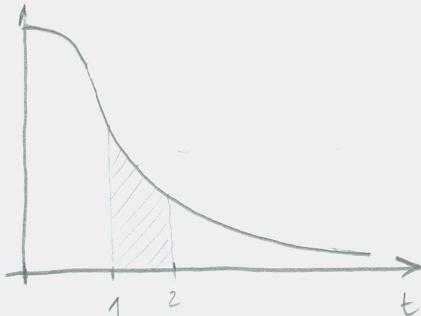
- ukupna površina ispod krivulje gustoće vjerojatnosti je 1

- aprimirasti - ULAZ: Poissonova raspodjela
- IZLIZ: eksponentijalna raspodjela

7.2. POVEZANOST POISSONOVE I EKSPONENTCIJALNE RASPODJELE

POISSONOV

- vrijeme između 2 uzastopnih dolaska u sustav



Vjerojatnost da će doći između 1. i 2. sekunde

BINOMNA RASPODJELA

- nije kontinuirana nego diskretna
- aproksimira Poissonovu

10

- primjer: koja je vjerojatnost da se u 10 bacanja kocke 3 puta pojavi 6-icu?

$$b(k; n, p) = b(3; 10, \frac{1}{6}) = \binom{10}{3} \left(\frac{1}{6}\right)^3 \left(\frac{5}{6}\right)^7$$

- Poissonova razdjeta: $n \rightarrow \infty$, $p \rightarrow 0$ i $vjerojatnost \rightarrow 0 \Rightarrow \lambda = np = konst.$

- ne treba znati izvode formula

- VJEROJATNOST da u jedinicnom periodu učini dogadaju: $e^{-\lambda}$

$$b(k; n, p) = \frac{\lambda^k}{k!} b(k-1; n, p)$$

$$\Rightarrow p(k; \lambda) = \frac{\lambda^k}{k!} e^{-\lambda}$$

... u jedinici vremena, uz zadani λ imamo k novih dolazaka

10



OBRAVITNE SUČAJNE VARIJABLE
= prosjek dolazaka novih poslova
u jedinici vremena (α)
 $\rightarrow \lambda$

- primjer: Poissonova raspodjela

kisa pada po Poissonovoj raspodjeli. Ako na neku površinu prosječno padne 100 kapi/m², koja je vjerojatnost da će u sljedećoj sekundi na tu površinu padne 2 kapi?

$$\alpha = \frac{100}{60} = 2$$

D-18

1 3

$$P(2; \lambda) = \frac{\lambda^2}{2!} e^{-\lambda} = 26,2\%$$

EKSPONENCIJALNA RAZDjOBA

7.3. ANALIZA SUSTAVJA S POISSONOVOM RAZDIOBOM DOLAZAKA I EKSPONENCIJALNOM RAZDIOBOM TRAJANJA OBRADE

- stokasticki sustav

$$-\alpha < \beta \rightarrow \frac{\alpha}{\beta} < 1$$

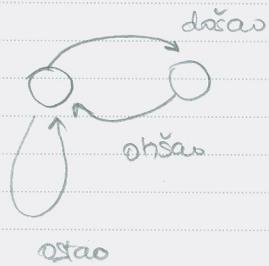
- ravnotežno stanje ($\alpha, \beta \dots$ kust.; sve vrijednosti su konstantne)

$\hat{z} \dots$ trenutni broj postrova u sustavu u trenutku t ; slučajna varijabla
 $\hat{n} \dots$ projekcija broj postrova u sustavu \Rightarrow tražimo

xDt ... vjerojatnost da je 1 posao došao
xDt ... " " " " otšao

HARKOVULDEU LANAC

-stavje i → toliko poglova imaju u sustavu
↳ vjerojatnosti da se nalazimo u tom stavju



slbroj = 1 ... KOMPLETAN MARKOVLODEV LANAC

$$p_i(t + \Delta t) = p_{i-1} \alpha \Delta t + p_{i+1} \beta \Delta t + p_i (\alpha + \beta) \Delta t$$

stavje i wažje
i i je desao

Staue l'v'e
i A otšan

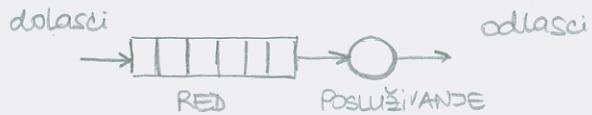
stavlja da niti jedan
nije došao ni otišao

- stacionarne stanje \Rightarrow derivacije su \neq 0?

ULAZ P izaz E za razliku
od Little uvijedi uvijek!

$$-p_0(t) = 1 - p \Rightarrow p_i(+) = (1-p)p^i \Rightarrow \bar{\pi}_e = \frac{p}{1-p} \Rightarrow \bar{T} = \frac{1}{\beta-\alpha}$$

ANALIZA VREMENSKIT SVOJSTAVA RAČUNALNITI SUSTAVA



- model sadar s 2 parametra:

- $\alpha [s^{-1}]$... broj dolazaka poslova u jedinici vremena
- $\beta [s^{-1}]$... broj poslova koje poslužitelj MOŽE obraditi u jedinici vremena
... ovise o snazi poslužitelja i vrsti poslova

- α, β ... dimenzija frekvencije

$1/\alpha$... prosječno vrijeme između 2 dolaska

$1/\beta$... prosječno trajanje posluživanja

- izlazne vrijednosti sustava:

- \bar{T} ... prosječno vrijeme zadržavanja poslova u sustavu
- \bar{n} ... prosječni broj poslova u sustavu

$$\boxed{\bar{n} = \alpha \bar{T}}$$
 LITTLEVO PRAVILO

• P ... faktor niskosti poslužitelja

$$\boxed{P = \frac{\alpha}{\beta}}$$

=> PRETPOSTAVKA :

- Uzaz: POISSON
- izlaz: EKSPONENCIJALNA

$$\boxed{\bar{n} = \frac{P}{1-P} = \frac{\alpha}{\beta-\alpha}}$$

$$\boxed{\bar{T} = \frac{1}{\beta-\alpha}}$$

$$q(i > N) = P^{N+1}$$

Zadatak

U nekom sustavu zabilježen je da se u jednoj godini pojavljuje Poissonova raspodjela uz $\lambda = 25^1$, a vrijeme obrade ima eksponentijalnu raspodjelu. Mjerenjem je utvrđeno da se u godini mijenja zadizavanja postala $T = 0,5$.

- o Koja je vjerojatnost da se u sustavu nalazi više od 5 poslova?

$$P(i > 5) = P^{5+1} = P^6$$

$$\bar{\mu} = \lambda T = 2 \cdot 0,5 = 1$$

$$\bar{\mu} = \frac{P}{1-P} \Rightarrow P = \frac{\bar{\mu}}{(\bar{\mu} + 1)} = 0,5 \Rightarrow 50\%$$

$$P(i > 5) = P^6 = 0,5^6 = 1,5625\%$$

- o Koja je vjerojatnost da se u sustavu nalazi između 2 i 4 posla?

$$\begin{aligned} P(2 \leq i \leq 4) &= P(i \geq 2) - P(i \geq 5) \\ &= P(i \geq 1) - P(i \geq 4) \\ &= P^{1+1} - P^{4+1} \\ &= P^2 - P^5 \\ &= 0,25 - 0,03125 \\ &= 0,21875 \end{aligned}$$