



Katedra za računarsku tehniku i informatiku





PERFORMANSE RAČUNARSKIH SISTEMA

Školska 2008/2009 godina

Nastavnici



- Predavanja: prof. Jelica Protić,
 - 2 časa nedeljno,
 - e-pošta: jeca@sezampro.yu
- Vežbe: Đorđe Jevđić,
 - 2 časa nedeljno
 - e-pošta: jevdjic@etf.rs
 - konsultacije po dogovoru
 - stranica predmeta: <http://rti.etf.rs/rti/prs>

Logistika



- Ispit: pismeni + domaći zadatak
- Broj poena se formira po formuli:
$$P = \max\{I, 0.7 \cdot I + 0.3 \cdot D\}$$
gde je I – broj poena osvojen na ispitu
D – broj poena osvojen na domaćem zadatku
- Skala ocena
 - $P \leq 50 \Rightarrow$ ocena 5
 - $50 < P \leq 60 \Rightarrow$ ocena 7
 - $60 < P \leq 70 \Rightarrow$ ocena 7
 - $70 < P \leq 80 \Rightarrow$ ocena 8
 - $80 < P \leq 90 \Rightarrow$ ocena 9
 - $90 < P \Rightarrow$ ocena 10

Literatura



- Materijali sa predavanja i vežbi
- *Jain Raj* – The Art of Computer Systems Performance Analysis Techniques for Experimental Design, Measurements Simulation and Modeling

Sadržaj



1. Uvod
2. Performanse procesora
3. Performanse operativne memorije (iskorišćenje)
4. Performanse operativne memorije (brzina)
5. Performanse diskova
6. Optičko memorisanje
7. Performanse magnetnih traka
8. Modelovanje performansi računarskih sistema

Sadržaj

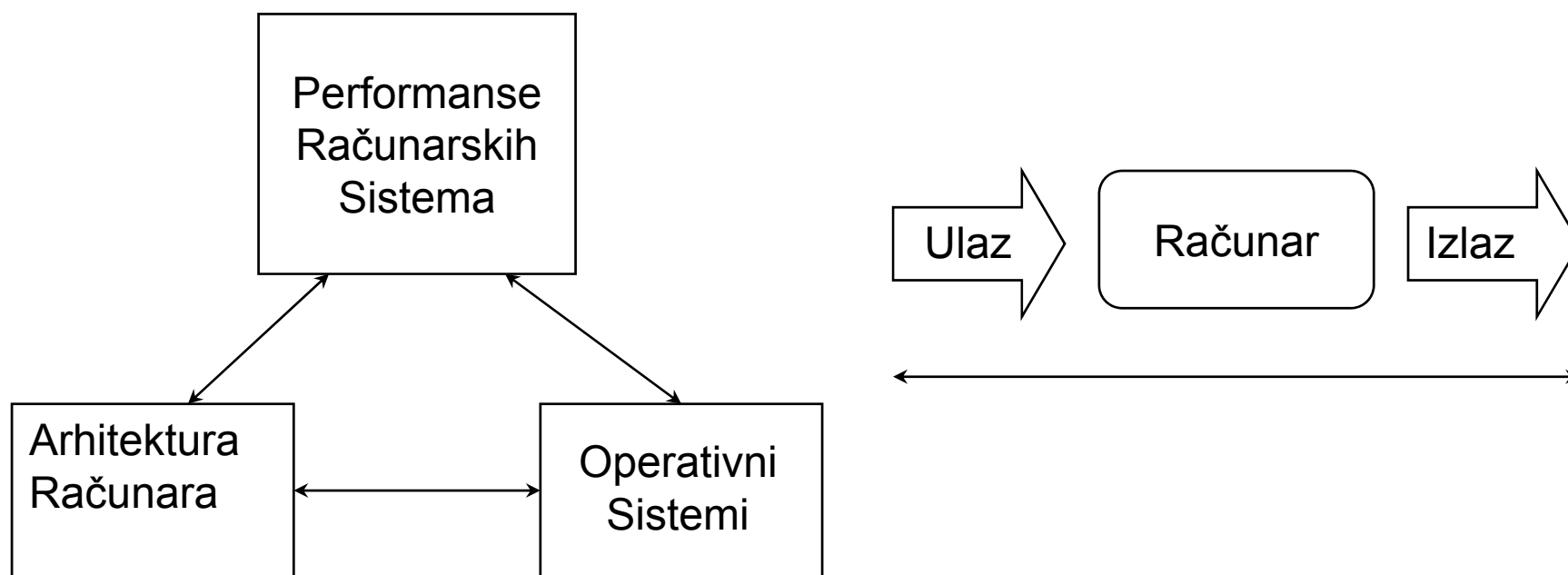


9. Poasonov proces (Simeon Denis Poisson, 1781-1840)
10. Neeksponencijalni modeli (SMO sa neeksponencijalnim raspodelama)
11. Paralelni serveri – teorijska razmatranja i modeli
12. Zatvorene mreže (definicija i analiza performansi)
13. Otvorene mreže SMO
14. Interaktivni sistemi
15. Operaciona analiza - Operacioni modeli računarskih sistema

1. Uvod



- Cilj izučavanja arhitekture računara i operativnih sistema jeste poboljšanje performansi računarskih sistema. Računar modeliramo kao dinamički sistem



1. Uvod



1. Performanse komponenti:

- procesora
- memorijskih organizacija
- ulazno-izlaznih uređaja
 - diskova, traka
 - štampača
 - monitora...

1. Uvod



2. Performanse sistema – osnovne teorije:

- teorija redova čekanja
- teorija sistema masovnog opsluživanja
- analiza performansi paketne obrade
- analiza interaktivnih sistema

3. Metodologije analize performansi:

- analitički modeli
- simulacioni modeli
- merenja

1. Uvod



Oblasti primene

Oblasti primene				
Projektovanje Računarskih Sistema	Izbor Računarskih Sistema	Upravljanje radom Računarskih Sistema		
***	*	**	Stohastički modeli	Analitičke tehnike
**	*	***	Operacioni modeli	
**	*	***	Specijalizovani računarski simulatori	Simulacione tehnike
**	*	**	Simulatori opšte namene	
-	**	***	Prirodna opterećenja	Merenja i merne tehnike
-	***	**	Veštačka opterećenja	
Predviđanje performansi	Poređenje performansi	Poboljšanje performansi		

Problemi

1. Uvod



- Najpre se vrši modelovanje, pa onda simulacija
- Stohastički modeli važe na beskonačnom vremenskom intervalu (slučajni brojevi, raspodele).
- Operacioni modeli su razvijeni specijalno za računarske sisteme. Važe u konačnom intervalu vremena, uz blaže uslove nego kod stohastičkih modela (učestanost, proporcije,...).
- Simulatori opšte namene GPSS (General Purpose Simulation Systems).
- Veštačka opterećenja – zasnovano na GSB (generator slučajnih brojeva).
- U principu se toleriše do 5% greške, mada se dozvoljava i do 40% za specijalne primene.

2. Performanse Procesora



Performanse zavise od arhitekture procesora.

Osnovni elementi arhitekture:

1. Registri

1.1 univerzalni (opšte namene)

1.2 registri podataka

1.3 registri za pokazivače na podatke

1.4 indeksni registri

...

2. Performanse Procesora



2. Mašinske instrukcije

2.1 broj i tip adresa po instrukciji

2.2 načini adresiranja

2.3 skup mašinskih instrukcija

2.4 fizički format instrukcija

2. Performanse Procesora



3. Numerička tačnost predstavljanja brojeva (celi, realni, ...)

4. Brzina procesora

- Na brzinu sa aspekta korisnika utiče mnogo faktora
- Merilo: vreme trajanja prosečne instrukcije
- Kakve su te instrukcije, šta je moguće uraditi jednom instrukcijom - > važno je sa aspekta problema koji programer treba da reši

2.1 Dužina programa u zavisnosti od adresnosti



- Posmatramo **m**-adresni procesor
- **Lk(m)**- dužina programa izražena u broju mašinskih instrukcija.
- Računanje izraza sa k argumenata zavisno od broja memorijskih adresa
- 0-adresni radi sa 2 operanda na vrhu steka i rezultat ostane na vrhu steka
- 1-adresni ostavlja rezultat u akumulatoru
- 2-adresni ostavlja rezultat na mestu jednog operanda (MM) ili u akumulatoru (MMA)
- 3-adresni - direktno radimo sa 3 memorijske lokacije

2.2 Merenje frekvencije pojave pojedinih inst.



■ Problem broja adresa po instrukciji:

		Formati instrukcija			
m	0	0	R	RR	RRR
	1	M(A)	MR(A)	MRR	-
	2	MM(A)	MMR	-	-
	3	MMM	-	-	-
		0	1	2	3
		r			

M-memorija, **R**-registar, **A**-Acc akumulator, sa zagradom opcija npr. M i M(A)

m - broj memorijskih adresa, **r**-broj registarskih operanada

0 - nema ni *m* ni *r*, **R**-jedna registarska adresa, **RR**-dve registarske adrese...

- **M**-jedna memorijska adresa, **A**-akumulator, **MR**-jedna memorijska adresa i jedna registarski operand
- **MRR**-jedna memorijska i dve registarske adrese
- **MMM**-tri memorijske adrese

2.2 Merenje frekvencije pojave pojedinih inst.



Statička frekvencija pojava naredbe – koliko se puta pojavljuje određena naredba u kodu (prosto prebrojavanje instrukcija u kodu)

Dinamička frekvencija pojava naredbe - ono što se stvarno izvrši (to se meri)

Posmatraćemo prvo statičku frekvenciju tipičnog programa – smatramo da tipičan program ima mnogo naredbi dodele vrednosti

2.2 Merenje frekvencije pojave pojedinih inst.



- Opšti oblik naredbe dodele vrednosti:
 $M_k := M_{k-1} * M_{k-2} * \dots * M_2 * M_1$ – opšti oblik aritmetičkog izraza (* je neka operacija)
- U programima ima 67% dodela vrednosti kod dinamičke frekvencije -Knuth 1971., kod statičkih frekvencija ima 41%-51% -Knuth 1975.(FORTRAN)
- Merili razni naučnici: Robinson, Torsun, Al Jaral, Elshoff, Kuck, Strehendt...
- Uopšte: 35%-50% svih instrukcija u programu predstavljaju dodele vrednosti.

2.2 Robinson & Torsun - raspodela



	k(broj operanada)	Pk
A=B	2	0.43
A=B+C	3	0.3
A=B+C+D	4	0.12
A=B+C+D+E	5+ (5 ili vise)	0.15

$$P_5=P_6=P_7= 0.05 \rightarrow P_5^+=P_5+P_6+P_7 = 0.15$$

Prosečna (očekivana) dužina instrukcije dodele vrednosti:

$$k= 2 \cdot P_2 + 3 \cdot P_3 + 4 \cdot P_4 + 5 \cdot P_5 + 6 \cdot P_6 + 7 \cdot P_7 = 3.14$$

2.1 Dužina programa u zavisnosti od adresnosti



- broj mašinskih naredbi za izračunavanje izraza M_k
- $M_k = M_1 * M_2 * M_3 \dots * M_{k-1}$ u zavisnosti od adresnosti:

m=0 (format 0)	m=1 (format MA)	m=2 (format MMA)	m=3 (format MMM)
PUSH M1	A=M1	A=M1*M2	T=M1*M2
PUSH M2	A=A*M2	A=A*M3	T=T*M3
*	A=A*M3	A=A*M4	T=T*M4
PUSH M3
*	A=A*M _{k-1}	A=A*M _{k-1}	T=T*M _{k-2}
POP M _k	M _k =A{store}	M _k =A {store}	M _k =T*M _{k-1}
2k-2	k	k-1	k-2

← Ukupan broj instrukcija $L_k(m)$

2.1 Dužina programa u zavisnosti od adresnosti



k	Pk	Lk(0)	Lk(1)	Lk(2)	Lk(3)
2	0.43	2	2	1	1
3	0.3	4	3	2	1
4	0.12	6	4	3	2
5	0.05	8	5	4	3
6	0.05	10	6	5	4
7	0.05	12	7	6	5

Matematičko očekivanje – $L(m) = P_k \cdot L_k$ sr.

$$L(m) = \sum_{k=2}^7 P_k \cdot L_k(m)$$

	4.28	3.14	2.14	1.57
	L(0)	L(1)	L(2)	L(3)

$$L(1) = k \approx \pi \quad L(0) = 1.36 \cdot k \approx (e/2) \cdot \pi$$

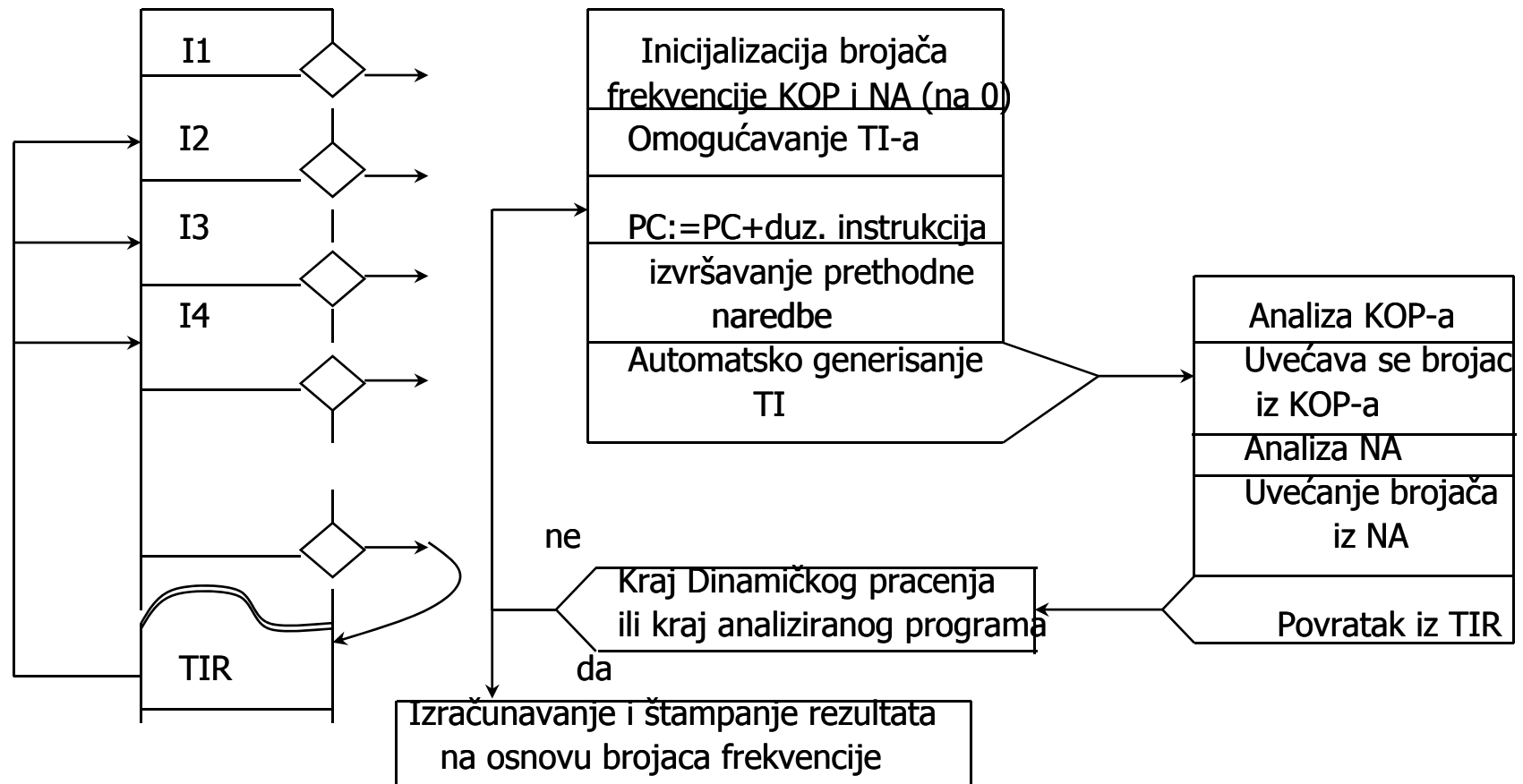
$$L(2) = 0.68 \cdot k \approx (e/4) \cdot \pi \quad L(3) = 0.5 \cdot k \approx \pi/2$$

2.3 Brzina procesora i vreme smese instrukcija T_{mix}



- Hoćemo da napišemo program koji dinamički prati rada procesora. Legenda:
- I – instrukcija
- TIR – trap interrupt routine
- Ukoliko je $T=1$ (trap bit), posle svake instrukcije se skače na TIR
- NA – način adresiranja
- KOP – kod operacije
- TI – trap interrupt – interni procesorski prekid

2.3 Brzina procesora i vreme smese instrukcija T_{mix}



2.3 Brzina procesora i vreme smese instrukcija T_{mix}



- Ako postoji konačan broj tipova naredbi i načina adresiranja:
- C_i - kodovi (svih) operacija: C_1, C_2, \dots, C_k
- A_k - načini adresiranja: A_1, A_2, \dots, A_k
- f_{ij} – broj (frekvencija) pojava kombinacija (C_i, A_j)
- t_{ij} - vreme izvršavanja instrukcija za svaku kombinaciju (C_i, A_j)
- T_{run} – **ukupno vreme izvršavanja** nekog programa
- F – **ukupna broj izvršenih instrukcija**

2.3 Brzina procesora i vreme smese instrukcija T_{mix}



$$T_{run} = \sum_{i=1}^n \sum_{j=1}^k f_{ij} \cdot t_{ij}$$

Vreme izvršavanja	T_{run}
Ukupan broj instrukcija	F

■ T_{mix} - vreme smese instrukcija (*instruction mix time*):

$$T_{mix} = T_{run} / F = \sum_{i=1}^n \sum_{j=1}^k p_{ij} \cdot t_{ij}$$

$$p_{ij} = f_{ij} / F$$

2.3 Brzina procesora i vreme smese instrukcija T_{mix}



- **Procesor se smatra bržim što je kraće vreme smese.**

- **Brzina procesora:**

$$V_p = 1/T_{mix} = F/T_{run} \quad [\text{instruction per sec}]$$

(izražava se u broju instrukcija u jedinici vremena) = $10^3/T_{mix}$ [KOPS] =
= $10^6/T_{mix}$ [MIPS]

- **KOPS** – kilo operacija u sekundi, **MIPS** – million instr. per second

$$V_p = 1 / \left(\sum_{i=1}^n \sum_{j=1}^k p_{ij} \cdot t_{ij} \right)$$

2.4 Kritike – ‘backronym’



- *Meaningless Indication of Processor Speed*
- *Meaningless Information on Performanse for Salespeople*
- *Meaningless Information about processor Speed*
- KIPS ili KOPS (10^3)
- MIPS (10^6)
- VAX 11/780 je bio 1 MIPS mašina
- GIPS – giga instructions per second (10^9)
- kMIPS – negde se koristi kilo mega
- zMIPS –koristio IBM za servere zSeries

2.3 Brzina procesora i vreme smese instrukcija T_{mix}



MIPS pojedinih procesora - Intel			
godina	procesor	frekvencija	MIPS
1978	8086	5 MHz	0.33
1982	80286	8 MHz	1.2
1988	80386DX	25 MHz	8.5
1999	Pentium III	500 MHz	1,354
2003	Pentium 4 Extreme Edition	3.2 GHz	9,726
2006	Intel Core 2 Extreme QX6700	3.33 GHz	57,063

- MIPS-ovi mogu da se porede kod procesora sa istim n i k , istom adresnošću i pod uslovom da se izvršava isti merni program