

# Multiprocesorski sistemi

## Laboratorijska vežba 4

### Snoopy protokoli

## Uvod

Cilj vežbe je da studenti korišćenjem simulatora poboljšaju poznavanje *snoopy* protokola za održavanje koherencije keš memorije u multiprocesorskim sistemima.

## Podešavanje okruženja

U programu za pregledanje Interneta uneti adresu <http://mups.etf.rs/simulatori/vivio/>. Na datoj adresi su dostupni simulatori za nekoliko protokola za koherenciju keš memorije. U slučaju da Vivio simulator nije dostupan na lokalnom računaru, ispratiti uputstva za podešavanje Vivio okruženja na lokalnom računaru i podesiti lokalni računar prema njima.

## Zadaci

### Poništavajući protokoli

Svaki od zadataka sadrži oznaku protokola na koji se odnosi kao prefiks u uglastim zagradama ispred postavke zadatka i upravo u tom simulatoru treba obaviti simulaciju. Preslikavanje adresa u keš memoriji je direktno u svim simulatorima.

1. [WTI] Izvršiti sledeću sekvencu operacija:

- |              |              |
|--------------|--------------|
| 1. P0, R, A0 | 3. P1, W, A0 |
| 2. P0, W, A0 | 4. P0, R, A0 |

Koliko puta je svaki procesor pristupao memoriji? Koji je hit rate svakog procesora za ovu sekvencu?

2. [WTI] Simulirani sistem koristi write-through strategiju upisa (svaki upis odmah ažurira i memoriju), a pri tom se koristi write-no-allocate mehanizam alociranja ulaza u kešu. Na osnovu čega se može zaključiti, posmatrajući prethodni primer, da se koristi baš taj mehanizam alokacije pri upisu? Kako bi izgledao nulti blok u kešu procesora P1, a kako blok A0 u memoriji posle ove sekvence, ako bi se koristio mehanizam write-allocate?

3. [WTI] Da li postoji razlika između ovog protokola i ponašanja keša u jednoprocesorskom sistemu pri čitanju i upisu? Ako da, opisati. Pretpostaviti da keš jednoprocesorskog sistema koristi isti mehanizam alokacije i istu strategiju upisa.

4. [WTI] Izvršiti sledeću sekvencu operacija:

- |              |              |              |
|--------------|--------------|--------------|
| 1. P0, R, A0 | 3. P2, W, A0 | 5. P0, W, A2 |
| 2. P1, W, A0 | 4. P0, R, A0 |              |

Posle izvršenja koraka 3, koji od procesorskih keševa ima validan podatak A0? Da li posle koraka 5 procesor ima neki validan podatak i koji? Da li je memorija ažurna?

5. [WTI] Izvršiti sledeću sekvencu operacija:

- |              |              |              |              |
|--------------|--------------|--------------|--------------|
| 1. P0, R, A0 | 3. P1, R, A0 | 5. P0, W, A0 | 7. P1, R, A0 |
| 2. P0, R, A0 | 4. P2, R, A0 | 6. P0, W, A0 | 8. P3, W, A0 |

Svih 8 operacija od strane sva četiri procesora pristupaju isključivo lokaciji A0. Koji od procesora imaju validnu vrednost tog podatka posle ove sekvence? Koja osobina ovog protokola utiče na loše performanse (u smislu nepotrebnih pristupa memoriji)? Kako je moguće ublažiti ove nedostatke?

6. [WTI] Ako se u primeru iz 4. pitanja poslednja operacija u sekvenci zameni sa P0, R, A2, šta se dešava prilikom zamene bloka u toj operaciji? Da li su potrebne posebne akcije protokola? Ako jesu, koje? Ako nisu, zašto?

7. [MSI] Izvršiti sledeću sekvencu operacija:

- |              |              |              |              |
|--------------|--------------|--------------|--------------|
| 1. P0, R, A0 | 3. P0, W, A0 | 5. P1, R, A0 | 7. P1, W, A2 |
| 2. P0, W, A0 | 4. P0, W, A0 | 6. P1, W, A0 | 8. P2, W, A2 |

Koliko puta je svaki procesor pristupao memoriji? Za svaki pristup navesti u kom koraku se odvija i razlog pristupa. Napomena: moguće je više pristupa memoriji od strane istog procesora tokom jedne operacije.

8. [MSI] Koja se suštinske razlike ovog protokola u odnosu na WTI uočavaju u koracima 1 i 8 prethodnog primera (u smislu načina upisa u keš memoriju)? Koja se prednost ovog protokola u odnosu na WTI uočava u koracima 3 i 4?
9. [MSI] Šta se dešava pri zameni bloka u koraku 7? Koje akcije protokola se odvijaju?
10. [MSI] Koja mana ovog protokola se uočava u koraku 2? Koja akcija protokola nije imala efekat?
11. [MESI] Izvršiti sekvencu operacija iz primera 7. Koja prednost se uočava u prvih nekoliko koraka?
12. [MESI] Izvršiti sledeću sekvencu operacija:
 

1. P0, R, A0	3. P2, R, A0	5. P1, R, A0	7. P2, W, A0
2. P1, R, A0	4. P2, W, A0	6. P1, R, A2	8. P0, W, A0

Koja se prednost uočava u koracima 2 i 3? Na osnovu kog signala se određuje u koje će stanje (E ili S) preći blok u keš memoriji u slučaju promašaja pri čitanju?
13. [MESI] Izvršiti sekvencu operacija iz prethodnog pitanja. U posle koraka 5, u kom se stanju nalazi podatak A0 u svim procesorima koji ga imaju kao validnog? Šta se dešava pri zameni u koraku 6? Koje se akcije protokola događaju?
14. [MESI] Zašto je blok A0 posle izvršavanja koraka 6. prešao u stanje S, a ne E, iako je to jedinstvena kopija u kešu procesora P2? Koja je mana protokola se samim tim manifestuje u koraku 7? Predložiti nadgradnju protokola koja bi rešila taj problem.
15. [MESI] U koraku 8, odakle se čita podatak A0 pre upisa? Da li se memorija pri tom ažurira i zašto?
16. [MOESI] Izvršiti sekvencu operacija iz 12. pitanja. U koraku 2, odakle se procesoru P1 dostavlja podatak A0? U koraku 3, odakle se procesoru P2 dostavlja podatak A0? Kako je to implementirano?
17. [MOESI] U kom stanju ostaje blok A0 u procesorima P1 i P2 posle koraka 5? Da li je memorija ažurna? Ako bi se između koraka 5 i 6 umetnuo korak P3, R, A0, od koga bi procesor P3 dobio podatak A0 i zašto?
18. [MOESI] Ako bi se umesto koraka 7 izvršila operacija P2, R, A2, koje bi bile akcije protokola pri zameni bloka?

## Ažurirajući protokoli

Svaki od zadataka sadrži oznaku protokola na koji se odnosi kao prefiks u uglastim zagradama ispred postavke zadatka i upravo u tom simulatoru treba obaviti simulaciju. Preslikavanje adresa u keš memoriji je direktno u svim simulatorima. Svi zadaci se odnose na sledeću sekvencu operacija:

- |              |              |              |              |
|--------------|--------------|--------------|--------------|
| 1. P0, R, A0 | 3. P2, R, A0 | 5. P1, R, A2 | 7. P0, R, A2 |
| 2. P1, R, A0 | 4. P0, W, A0 | 6. P1, W, A2 | 8. P0, W, A2 |

19. [DRAGON] Koliko puta je svaki procesor pristupao memoriji? Koji je hit rate svakog procesora za ovu sekvencu?
20. [DRAGON] U koje stanje prelazi blok sa podatkom A0 u kešu procesora P0 posle koraka 2. i 3? Odakle se dostavlja taj podatak procesorima P1 i P2 i u koje stanje prelazi odgovarajući blok u tim procesorima?
21. [DRAGON] U koje stanje prelazi blok sa podatkom A0 posle koraka 4, u svim procesorima koji imaju taj podatak i koje su vrednosti tog podatka u tim procesorima? Da li se ažurira memorija? Koja je suštinska razlika ovog protokola u odnosu na sve prethodno obrađene vidljiva već u ovom koraku?
22. [DRAGON] U kom stanju će biti blok sa podatkom A2 u procesoru P1 posle čitanja u koraku 5, a u kom stanju posle upisa u koraku 6? Zašto blok nije prešao u stanje Sm? Na osnovu čega se određuje stanje u koje će preći? Koje su bile akcije protokola pri zameni?
23. [DRAGON] Koje su akcije protokola u koraku 7? Šta se dešava pri zameni i zašto? Odakle se dostavlja podatak A2 procesoru P0? Da li je memorija ažurna što se tiče podataka A0 i A2? U kojim stanjima se nalazi podatak A2 u svim procesorima koji ga imaju?
24. [DRAGON] Koje su akcije protokola u koraku 8? U kom stanju će se naći blokovi sa podatkom A2 u procesorima koji imaju taj podatak? Da li su sve kopije ažurne?
25. [DRAGON] Ako bi se posle navedene sekvence izvršila operacija 9. P2, W, A2, opisati sve akcije protokola koje bi usledile. U kom stanju će se naći blokovi sa podatkom A2 u svim procesorima koji imaju taj podatak? Koji sve procesori su pristupali memoriji u ovom koraku i iz kog razloga? Da li je memorija ažurna?
26. [FIREFLY] U kom stanju se nalazi podatak A0 u procesoru P0 posle koraka 1? Šta označava to stanje i koji je ekvivalent tom stanju u DRAGON protokolu?

27. [FIREFLY] U kom stanju se nalazi podatak A0 u procesorima P0 i P1 posle koraka 2? Koji je ekvivalent tom stanju u DRAGON protokolu? Odakle procesor P1 dobija podatak?
28. [FIREFLY] U kom stanju se nalazi podatak A0 u procesorima P0, P1 i P2 posle koraka 3? Odakle procesor P2 dobija podatak?
29. [FIREFLY] U kom stanju se nalazi blok sa podatkom A0 u svim procesorima posle koraka 4? Da li je memorija ažurna? Koja dva stanja DRAGON protokola objedinjuje ovo stanje? Zbog čega je moguće objediniti ova dva stanja u jedno?
30. [FIREFLY] Da li su potrebne akcije protokola pri zameni u koraku 5? U koje stanje prelazi blok A2 u procesoru P1 posle koraka 6? Šta označava to stanje i koji je ekvivalent tom stanju u DRAGON protokolu?
31. [FIREFLY] Koje su akcije protokola u koraku 7? Šta se dešava pri zameni i zašto? Odakle se dostavlja podatak A2 procesoru P0? Da li je memorija ažurna što se tiče podataka A0 i A2? U kojim stanjima se nalazi podatak A2 u svim procesorima koji ga imaju?

**VAŽNO:** Ukoliko u bilo kom zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku i da nastavi da izgrađuje svoje rešenje temeljima uvedene pretpostavke.