

ARH2 – 2. BLIC

1. Za vektorske funkcije raspršenja vrijedi:
 - **spremaju više skalara na memorijske lokacije koje ne moraju biti uzastopne**
2. Kako vektoriziranje algoritma tipično utječe na učestalost hazarda?
 - **povećava broj hazarda vrste RAW**
3. Koji od sljedećih pojmova ne označava vrstu funkcijskog paralelizma?
 - **paralelizam u vektorskoj funkcijskoj jedinici**
4. Koji glavni učinak na izvođenje programa možemo očekivati nakon vektorizacije algoritma?
 - **smanjivanje broja izvedenih instrukcija**
5. Koji od sljedećih pojmova ne označava ovisnost koja može usporiti obradu u procesoru s dinamičkim izdavanjem?
 - **glagolska ovisnost**
6. Za vektorske i skalarne instrukcije tipično vrijedi:
 - **izvode se u različitim sklopovskim jedinicama i koriste različite registerske skupove**
7. Koja je razlika između dretve i procesa?
 - **dretve imaju zajednički adresni prostor, a procesi ne**
8. Paralelizam na razini programskih petlji je klasificiran kao:
 - **srednje zrnati**
9. Za vektorske instrukcije maskiranja vrijedi da:
 - **iz dva vektorska operanda proizvode vektorski operand**
10. Faktor ubrzanja za istodobno izdavanje najviše 15 instrukcija iznosi:
 - **oko 3**
11. Koja je razlika između imenske i podatkovne ovisnosti?
 - **Imensku ovisnost možemo izbjeći povećanjem broja registara**
12. Iscrpne analize ispitivanja programa pokazuju da se najčešće mogu usporedno izvršavati:
 - **3 instrukcije**
13. Koji od navedenih pojmova ne predstavlja jednu od 6 glavnih vrsta vektorskih instrukcija?
 - **instrukcije kontrakcije / instrukcije indikacije**
14. Koji od navedenih pojmova nije povezan s imenskom ovisnošću?
 - **podatkovna ovisnost**
15. Pojavu koja uzrokuje zastoj protočne arhitekture nazivamo:
 - **hazardom**
16. IA-64 je porodica Intelovih procesora sa arhitekturom:
 - **EPIC**
17. Koncept EPIC odnosi se na:
 - **eksplicitno izražen paralelizam na razini instrukcija**
18. Što od navedenog ne spada u iskorišteni funkcijski paralelizam?
 - **podatkovni paralelizam**
19. Jedna od glavnih prednosti višerazinske stranične tablice:
 - **manja cijena neiskorištenih dijelova LAP-a**
20. Vrste podatkovnog hazarda:
 - **WAR,RAW,WAW**
21. Višetračna vektorska obrada podrazumijeva:
 - **višestruke funkcijske jedinice**
22. Kod straničenja adresno preslikavanje izvodi se pomoću sljedeće operacije:
 - **prozivanje tablice**
23. Kod segmentacije adresno preslikavanje izvodi se pomoću sljedeće operacije:
 - **zbiranje**
24. Najčešći odnos TLB i PM.
 - **TLB prethodi PM**

25. Za vektorske instrukcije redukcije vrijedi da:
 - **iz vektorskih operanada proizvode skalarni operand**
26. Paralelizam na razini dretvi i procesa isključivo se koristi:
 - **na razini kombinacije arhitekture i operacijskog sustava procesora**
27. Disipacija snage procesora ovisi:
 - **linearno o frekvenciji signala vremenskog izvođenja**
28. Performansa procesora veća je ako je:
 - **manji prosječan broj perioda po instrukciji CPI**
29. Neka je zadano računalo sa stranicama od 4KB. Koliko će najmanje fizičkog RAM-a zauzeti proces koji koristi ukupno 4097 bajtova memorije?
 - **8192B**
30. Kakav sve odnos između LAP i FAP se može pronaći na postojećim 32-bitnim računalima?
 - **bilo kakav; $LAP < FAP$, $LAP > FAP$ ili $LAP = FAP$**
31. Kako se kod straničenja rješava problem veličine stranične tablice?
 - **višerazinskom organizacijom**
32. Neka je zadano 32-bitno računalo s 1 GB memorije RAM-a, 1 MB PML2 te 16KB PML1. Koliko je FAP?
 - **2^{30}**
33. Neka je zadano 32-bitno računalo s 1 GB memorije RAM-a, 1 MB PML2 te 16KB PML1. Koliko je LAP?
 - **2^{32}**
34. Znatost zaštite pristupa kod straničenja je:
 - **na razini stranice**
35. Ako zanemarimo promašaje stranice, vrijeme pristupa memorijskom podatku na modernom računalu je:
 - **varira puno zbog mogućih promašaja u PM i TLB**
36. Jedan od nedostataka straničenja je:
 - **unutrašnja fragmentacija**
37. Kad se događa iznimka pogreške stranice?
 - **kad god se adresira stranica koja nije prisutna u straničnoj tablici procesa**
38. Zašto je $LAP > FAP$ poželjniji od odnosa $LAP = FAP$?
 - **zbog mogućnosti proširenja**
39. Za superskalarne RISC arhitekture je specifično da se paralelno izvođenje slijednog programa pspješuje:
 - **dinamičkom analizom međuovisnosti instrukcija u sklopovskom procesoru (scoreboard)**
40. Koncept protočnosti je koristan jer omogućava:
 - **iskorištavanje instrukcijskog paralelizma**
41. Slijed instrukcija load r5, 20(r1); add r2, r1, r5 može rezultirati:
 - **hazardom tipa RAW**
42. Translacijski spremnik ne sadrži:
 - **kopiju bloka podataka iz RAM-a**
43. Translacijski spremnik sadrži:
 - **kopije straničnih opisnika**
44. Zaokruži ispravan redosljed memorijske hijerarhije:
 - **registri, cache, RAM, disk**
45. Kako se kod straničenja rješava problem brzine pristupa elementima stranične tablice:
 - **cacheiranjem straničnih opisnika**
46. U kontekstu IA-64 arhitekture i1 || i2 & i3 znači:
 - **prve dvije paraleleno i zadnja slijedno.**
47. Što omogućavamo preimenovanjem registra:
 - **otklanjanje hazarda WAR**

48. Utjecaj podatkovnih hazarda RAW na performansu računala ne može se ublažiti:
- **internim prosljeđivanjem rezultata**
49. Kao glavni učinak na izvođenje programa možemo očekivati nakon vektoriziranja algoritama:
- **smanjivanje broja izvedenih instrukcija**
50. Odnos latencije i takta procesora.
- **1:100**
51. Za RISC procesore kritičan hazard je:
- **RAW**
52. U protočnom računalu sa zajedničkom jednoadresnom priručnom memorijom podataka i instrukcija naročito možemo očekivati:
- **strukturne hazarde**
53. Moderni superskalarni procesori tipično postižu:
- **CPI < 1**
54. Ako se logički i fizički adresni prostor podijele na jednake stalne veličine govorimo o virtualnom memorijskom sustavu sa:
- **straničenjem**
55. Zašto je uvjetni registar posebno problematičan na superskalarnim računalima:
- **postaje implicitni izvor podatkovnog hazarda**