

Ispitna pitanja

1. Šta je operativni sistem? Koje su komponente računarskog sistema i koja je veza operativnog sistema sa ostalim komponentama?
2. Koje su tri osnovne uloge operativnog sistema? Objasniti ukratko svaku od njih.
3. Koji su kriterijumi za poređenje operativnih sistema? Ukratko opisati svaki od njih.
4. Opisati korake koji se dešavaju nakon pokretanja računara. Šta se dešava nakon što je sistem inicijalizovan? Prikazati vremenski dijagram koji pokazuje kombinovano izvršavanje korisničkih procesa i prekidnih rutina.
5. Objasniti strukturu ulazno/izlaznog (U/I) podsistema, sa stanovišta softvera i hardvera. Šta je DMA i zašto je on značajan?
6. Struktura operativnog sistema. Koja je razlika multi-programiranog sistema i sistema koji obezbeđuje deljenje vremena (multitasking)?
7. Šta je dualni mod izvršavanja i zašto je neophodan? U kojim slučajevima se vrši prelazak iz korisničkog u kernel režim rada, a u kojim iz kernel režima u korisnički?
8. Šta je proces? U kojim stanjima može da se nađe proces i pod kojim uslovima prelazi iz jednog stanja u drugo? Nacrtati dijagram stanja procesa.
9. Redovi raspoređivanja procesa. Koji redovi procesa postoje u sistemu? Objasniti i prikazati dijagram čekanja procesa.
10. Šta su planeri i koje vrste planera postoje u sistemu? Šta su CPU-kontrolisani, a šta U/I-kontrolisani procesi? O čemu planeri moraju da vode računa u vezi sa ova dva tipa procesa? Zbog čega se na nekim sistemima uvodi i srednjeročni planer?
11. Koje se operacije mogu vršiti na procesima? Objasniti kako se kreiraju novi procesi u Linux operativnom sistemu. Objasniti sistemske pozive *fork* i *exec*.
12. Koja dva modela interprocesne komunikacije postoje? Koje su im prednosti i mane, a u kojim situacijama mogu da se koriste?
13. Deljena memorija kao mehanizam za interprocesnu komunikaciju. Prikazati primer rešenja proizvođač-potrošač problema korišćenjem deljene memorije i ograničenog bafera.
14. Prenošenje poruka kao mehanizam za interprocesnu komunikaciju. Koje metode za logičku implementaciju veze i operacija send/receive postoje? Objasniti ukratko svaku od njih.
15. Komunikacija u klijent-server sistemima. Objasniti utičnice, navesti koje vrste utičnica postoje i opisati šta je neophodno da se ispuni da bi se klijent i server povezali korišćenjem utičnica.
16. Objasniti poziv udaljenih procedura (RPC) kao primer klijent-server komunikacionog modela. Šta su posrednici i koja im je uloga? Kako je obezbeđeno izvršavanje udaljene procedure "tačno jednom"? Koja je uloga *randevouz* (matchmaker) servisa?
17. Objasniti cevi kao primer klijent-server komunikacionog modela. Koje vrste cevi postoje? Objasniti svaku od njih kao i način na koji se koriste.
18. Šta su niti i koje su prednosti programa koji koriste višestruke niti izvršavanja?
19. Amdalov zakon. Po čemu se razlikuju konkurentni i paralelni sistemi?
20. Koji su izazovi kod programiranja višejezgarnih sistema? Koji tipovi paralelizma postoje?
21. Koji modeli višestrukih niti postoje, sa stanovišta veze korisničkih i kernel niti? Objasniti ukratko svaki od modela.
22. Koji se problemi javljaju prilikom rada sa višestrukim nitima? Ukratko objasniti svaki od problema.
23. Objasniti problem kritične sekcije na primeru istovremene modifikacije deljene promenljive od strane dva procesa. Koja tri zahteva mora da zadovoljava svako rešenje problema kritične sekcije?

24. Prikazati Patersonov algoritam na primeru sa dva procesa. Dokazati korektnost rešenja problema kritične sekcije primenom Patersonovog algoritma dokazom uzajamne isključivosti, napretka i ograničenog čekanja.
25. Hardverska podrška za sinhronizaciju. Prikazati rešenje problema kritične sekcije korišćenjem `test_and_set` instrukcije.
26. Šta su muteksi i kako se koriste za zaštitu kritične sekcije. Koje su njihove prednosti i mane?
27. Šta su semafori i kako se koriste za zaštitu kritične sekcije. Koje je poboljšanje uvedeno kod semafora u odnosu na mutekse? Objasniti implementaciju semafora i ulogu liste blokiranih procesa.
28. Rešenje problema proizvođač-potrošač sa ograničenim baferom korišćenjem semafora.
29. Rešenje prvog problema čitača/pisača korišćenjem semafora (prednost čitačima).
30. Rešenje drugog problema čitača/pisača korišćenjem semafora (prednost pisačima).
31. Prisvojivo i neprisvojivo raspoređivanje. U koje četiri situacije se donose odluke u vezi sa raspoređivanjem procesora i kakav to uticaj ima na tip raspoređivanja (prisvojivo-nepisvojivo)? Prednosti i mane oba pristupa.
32. Na primeru sistema sa više procesa prikazati FCFS algoritam raspoređivanja procesora. Koje su prednosti a koje mane ovog pristupa.
33. Na primeru sistema sa više procesa prikazati prisvojivu i neprisvojivu verziju SJF algoritma raspoređivanja procesora. Kako se rešava problem u vezi sa nemogućnošću poznavanja vremena izvršavanja narednog CPU ciklusa za dati proces?
34. Na primeru sistema sa više procesa prikazati prisvojivu i neprisvojivu verziju algoritma prioritetnog raspoređivanja procesora. Koje su prednosti a koje mane ovog pristupa.
35. Na primeru sistema sa više procesa prikazati prisvojivu i neprisvojivu verziju Round-Robin (RR) algoritma raspoređivanja procesora. Šta je vremenski kvant i na koji način odabir vremenskog kvanta utiče na RR algoritam raspoređivanja?
36. Objasniti na primeru dinamičko raspoređivanje u višestrukim redovima. Na osnovu kojih parametara se definiše planer koji koristi ovakav algoritam raspoređivanja?
37. Koji pristupi postoje kod raspoređivanja u multiprocesorskim sistemima? Šta je afinitet procesora, a šta balansiranje opterećenja?
38. U kom smislu se algoritam za raspoređivanje sa monotonom stopom smatra optimalnim? Koje ograničenje on ima u pogledu maksimalne iskorišćenosti procesora? Uporediti na primeru ovaj algoritam sa EDF algoritmom raspoređivanja.
39. Šta je keš memorija? Koja je osnovna ideja, a koji ishodi pokušaja pristupa nekoj memorijskoj lokaciji, u zavisnosti od toga da li se ona nalazi u keš memoriji?
40. Prikazati osnovne algoritme keširanja (čitanje i upis).
41. Šta je direktno mapiran keš. Za proizvoljan primer veličine keša, nacrtati sliku i objasniti. Kako veličina bloka kod direktno mapiranog keša utiče na performanse i kako se određuje optimalna veličina bloka.
42. Šta su konfliktni promašaji kod direktno mapiranog keša. Dati primer i objasniti kako se taj problem razrešava?
43. Objasniti potpuno asocijativno keširanje i N-grupa set asocijativno keširanje. Koje su prednosti a koje mane u ova dva pristupa.
44. Objasniti *write-back* algoritam zamene kod upisa u keš. Prikazati modifikovane osnovne algoritme keširanja uz korišćenje *Dirty* bit indikatora.

45. Šta je kontinualna alokacija memorije, koji registri se koriste kod ovog pristupa i na koji način se vrši zaštita memorije kod ovog pristupa? U čemu je razlika između particionisanja sa fiksnom i promenljivom veličinom particija?
46. Koji algoritmi se koriste kod kontinualnog alociranja memorije? Uporediti ih. Šta je eksterna, a šta interna fragmentacija. Dati primer. Na koji način se može rešavati problem eksterne fragmentacije?
47. Šta je segmentacija i kako izgleda hardver za segmentaciju? Koje su prednosti a koje mane ovakvog pristupa upravljanju memorijom?
48. Šta je straničenje i koje su prednosti sistema koji implementiraju straničenje kao mehanizam upravljanja memorijom? Objasniti osnovni princip tumačenja logičke adrese kod straničenja i prikazati hardver koji omogućava korišćenje stranica u sistemu.
49. Kako izgleda hardverska podrška u sistemima koji implementiraju straničenje i zbog čega je ona neophodna? Objasniti upotrebu bafera za keširanje translacija (TLB) i prikazati blok dijagram sistema za translaciju logičkih u fizičke adrese koji koristi TLB. Šta je identifikator adresnog prostora (ASID) i šta nam on omogućava?
50. Na koji način se vrši zaštita memorije u sistemima koji koriste stranice? Na koji način je omogućeno deljenje stranica i šta time dobijamo?
51. Za dati virtualni i fizički adresni prostor, sadržaj TLB-a i tabele stranica, odgovoriti na pitanja slično kao u primeru sa ppt slajdova.
52. Koje različite strukture tabele stranica postoje? Ukratko opisati svaku od njih.
53. Šta su stranice na zahtev, šta podrazumeva ovaj koncept i kako se bit validnosti koristi u tu svrhu? Šta je greška stranice i kako izgleda rutina za obradu izuzetka greške stranice?
54. Performanse sistema sa stranicama na zahtev. Koji su najkritičniji koraci izazvani greškom stranice i kolika verovatnoća greške stranice je prihvatljiva da bi sistem imao prihvatljivu degradaciju performansi?
55. Na koji način može doći do prekomerne alokacije memorije korišćenjem stranica na zahtev? Kako zamena stranica pomaže u ovom scenariju? Kako se modifikuje rutina za obradu izuzetka usled greške stranice u slučaju da koristimo zamenu stranica?
56. FIFO algoritam zamene stranica? Za dati referentni niz i broj slobodnih okvira, pokazati kako ovaj algoritam zamenjuje stranice. Koje su prednosti a koje mane ovog algoritma? Šta je Beladijeva anomalija?
57. LRU algoritam zamene stranica. Koje dve implementacije postoje i zašto su problematične? Na koji način se može aproksimirati LRU algoritam?
58. Na koji način se alociraju memorijski okviri? Čime je određen minimalan broj okvira datog procesa i koji algoritmi postoje za alokaciju memorijskih okvira?
59. Šta je preopterećenost virtualnog adresnog prostora i zbog čega nastaje? Na koji se način rešava taj problem korišćenjem modela radnog skupa, a na koji način praćenjem učestalosti grešaka stranica?
60. Šta su memorijski mapirane datoteke? Koje su prednosti tog pristupa? Koji model među-procesne komunikacije se može implementirati korišćenjem memorijski mapiranih datoteka i na koji način?
61. Po čemu se alokacija kernel memorije razlikuje od alokacije procesa u korisničkom režimu? Koja dva algoritma se najčešće koriste za alokaciju kernel memorije? Koje su prednosti i mane svakog od njih?