**50 Pitanja i odgovora za prezentaciju "Pregled Računalnog Sustava"**

1. **Što je operacijski sustav?**
	* **OS upravlja hardverskim resursima, pruža usluge korisnicima i upravlja memorijom i I/O uređajima.**
2. **Koji su osnovni elementi računalnog sustava?**
	* **Procesor, glavna memorija, ulazno/izlazni moduli i sabirnica.**
3. **Koja je uloga procesora?**
	* **Kontrolira rad računala i izvršava funkcije obrade podataka.**
4. **Što je glavna memorija?**
	* **Memorija koja pohranjuje podatke i programe tijekom rada, poznata kao primarna memorija.**
5. **Koja je funkcija ulazno/izlaznih modula?**
	* **Premještaju podatke između računala i vanjskog okruženja.**
6. **Što omogućuje sabirnica?**
	* **Komunikaciju između procesora, glavne memorije i I/O modula.**
7. **Što je mikroprocesor?**
	* **Procesor na jednom čipu, ključna komponenta modernih računala.**
8. **Koja je uloga GPU-a?**
	* **Optimizirano računanje za grafičke i numeričke zadatke poput simulacija i velikih proračuna.**
9. **Što su DSP procesori?**
	* **Procesori za obradu audio i video signala te sigurnosne funkcije.**
10. **Što je SoC (Sustav na čipu)?**
	* **Čip koji integrira CPU, GPU, DSP, memoriju i I/O uređaje.**
11. **Što je ciklus instrukcija?**
	* **Proces dohvata i izvršenja instrukcija pohranjenih u memoriji.**
12. **Što radi registar instrukcija (IR)?**
	* **Pohranjuje trenutnu instrukciju koju procesor izvršava.**
13. **Koji su tipovi prekida?**
	* **Programski, tajmerski, I/O i hardverski.**
14. **Zašto su prekidi važni?**
	* **Povećavaju učinkovitost procesora omogućujući njegovo korištenje dok uređaji čekaju.**
15. **Što je višenitno izvršavanje?**
	* **Izvođenje više zadataka istovremeno korištenjem više niti ili procesora.**
16. **Koji su pristupi upravljanju višestrukim prekidima?**
	* **Onemogućavanje prekida tijekom obrade ili korištenje prioriteta.**
17. **Što je hijerarhija memorije?**
	* **Organizacija memorije prema brzini, cijeni i kapacitetu.**
18. **Što je princip lokalnosti u memoriji?**
	* **Memorijske reference grupiraju se na određene lokacije, što optimizira performanse.**
19. **Koja je uloga predmemorije (cache)?**
	* **Povećava brzinu procesora privremenim pohranjivanjem često korištenih podataka.**
20. **Što je L1, L2 i L3 cache?**
	* **Razine predmemorije s različitim brzinama i kapacitetima.**
21. **Koje su glavne kategorije predmemorije?**
	* **Veličina, blokovi, funkcije mapiranja, algoritmi zamjene i politike pisanja.**
22. **Koji je algoritam zamjene u predmemoriji najčešći?**
	* **Najmanje nedavno korišten (LRU).**
23. **Što je DMA (Izravni pristup memoriji)?**
	* **Omogućuje prijenos podataka između memorije i uređaja bez intervencije procesora.**
24. **Koja je prednost DMA-a?**
	* **Smanjuje opterećenje procesora tijekom prijenosa podataka.**
25. **Koje su tehnike I/O operacija?**
	* **Programirani I/O, I/O vođen prekidima i DMA.**
26. **Što su simetrični multiprocesori (SMP)?**
	* **Sustavi s više procesora koji dijele memoriju i I/O uređaje.**
27. **Koje su prednosti SMP-a?**
	* **Veće performanse, dostupnost, skalabilnost i rast sustava.**
28. **Što je višejezgreni procesor?**
	* **Čip s više procesorskih jezgri na jednom komadu silicija.**
29. **Kako se koristi GPU za opće zadatke?**
	* **Za obavljanje masovnih paralelnih izračuna poput simulacija i analiza.**
30. **Što je predmemorijska politika pisanja?**
	* **Određuje kada se promjene u predmemoriji ažuriraju u glavnoj memoriji.**
31. **Što je magnetska traka?**
	* **Sekundarna memorija korištena za pohranu velikih količina podataka s niskom cijenom.**
32. **Kako sabirnica podržava komunikaciju?**
	* **Omogućuje prijenos podataka, instrukcija i signala između komponenata sustava.**
33. **Što je glavna memorija u hijerarhiji memorije?**
	* **Primarna razina koja pohranjuje podatke i instrukcije aktivne tijekom obrade.**
34. **Kako funkcionira DMA prijenos?**
	* **Prenosi cijeli blok podataka iz memorije u uređaj bez procesora.**
35. **Što su ulazno/izlazni moduli?**
	* **Upravljački moduli za komunikaciju između računala i vanjskog svijeta.**
36. **Kako prekidi poboljšavaju učinkovitost?**
	* **Omogućuju procesoru da radi na drugim zadacima dok čeka I/O operacije.**
37. **Koja je razlika između programiranog I/O i DMA-a?**
	* **Programirani I/O zahtijeva stalnu pažnju procesora, dok DMA omogućuje autonomne prijenose.**
38. **Koje su funkcije procesora tijekom prekida?**
	* **Pohranjuje stanje, izvršava obradu i vraća kontrolu programu.**
39. **Kako se upravlja višestrukim prekidima?**
	* **Korištenjem prioriteta ili privremenim onemogućavanjem prekida.**
40. **Što je sekundarna memorija?**
	* **Memorija poput diskova i traka koja pohranjuje podatke trajno.**
41. **Što je predmemorijski algoritam najmanje korištenog bloka?**
	* **Zamjenjuje blok koji je najdulje bio nekorišten u predmemoriji.**
42. **Koja je svrha registara procesora?**
	* **Pohranjuju privremene podatke i upravljačke informacije tijekom obrade.**
43. **Kako hijerarhija memorije utječe na performanse?**
	* **Omogućuje brzi pristup kritičnim podacima smanjujući učestalost sporijih pristupa.**
44. **Koji su izazovi u dizajnu predmemorije?**
	* **Veličina predmemorije, funkcija mapiranja i algoritmi zamjene.**
45. **Što je vremenski ciklus instrukcije?**
	* **Vrijeme potrebno za dohvaćanje i izvršavanje jedne instrukcije.**
46. **Što omogućuje preemptivni prekid?**
	* **Prekida trenutni zadatak za obradu hitnijih događaja.**
47. **Koji uređaji koriste I/O module?**
	* **Diskovi, pisači, mrežni adapteri i drugi periferijski uređaji.**
48. **Koji su izazovi višeprocesorske organizacije?**
	* **Sinkronizacija, dijeljenje resursa i skalabilnost.**
49. **Što je zapisivanje s odgodom u predmemoriji?**
	* **Promjene se zapisuju u glavnu memoriju samo kada se blok zamijeni.**
50. **Kako se koriste višezadaci u modernim procesorima?**
	* **Korištenjem više niti i jezgri za paralelnu obradu zadataka.**

**50 Pitanja i odgovora za prezentaciju "Pregled operacijskih sustava"**

1. **Što je operacijski sustav?**
	* Operacijski sustav (OS) je softver koji upravlja hardverom računala i omogućava izvođenje aplikacija.
2. **Koji su osnovni ciljevi operacijskog sustava?**
	* Pogodnost, učinkovitost i razvoj.
3. **Koje usluge pruža operacijski sustav?**
	* Izvršenje programa, pristup I/O uređajima, kontrola datoteka, otkrivanje grešaka i praćenje aktivnosti.
4. **Što znači da je OS upravitelj resursima?**
	* OS upravlja resursima poput procesora, memorije i I/O uređaja, omogućavajući njihov optimalni rad.
5. **Što je sučelje za programiranje aplikacija (API)?**
	* API omogućuje aplikacijama pristup funkcijama OS-a i korištenje hardverskih resursa.
6. **Koje su osnovne faze evolucije operacijskih sustava?**
	* Serijska obrada, jednostavni sljedni sustavi, multiprogramiranje i sustavi dijeljenja vremena.
7. **Što je serijska obrada?**
	* Najraniji sustavi bez OS-a, gdje korisnici izravno upravljaju računalnim resursima.
8. **Koji su problemi serijske obrade?**
	* Gubitak vremena pri zakazivanju i postavljanju programa.
9. **Što je jednostavni sljedni sustav?**
	* Sustav s OS-om koji kontrolira redoslijed poslova bez izravnog pristupa korisnika procesoru.
10. **Što su multiprogramirani paketni sustavi?**
	* Sustavi koji omogućuju paralelno izvršavanje više programa koristeći multiprogramiranje.
11. **Što je sustav dijeljenja vremena?**
	* OS dijeli procesorsko vrijeme među više korisnika, omogućujući brzi odgovor svakom korisniku.
12. **Koje su komponente OS-a?**
	* Upravitelji za procese, memoriju, datoteke, uređaje i sigurnost.
13. **Koje su ključne funkcije upravitelja memorije?**
	* Alokacija memorije, izolacija procesa i zaštita podataka.
14. **Što znači izolacija procesa?**
	* Svaki proces ima vlastiti memorijski prostor, čime se sprječavaju neželjene interakcije.
15. **Što je straničenje?**
	* Tehnika koja omogućava da se proces sastoji od blokova fiksne veličine, stranica.
16. **Kako funkcionira straničenje?**
	* Proces koristi virtualne adrese koje se preslikavaju u fizičke adrese pomoću tablica stranica.
17. **Koja su glavna sučelja OS-a?**
	* ISA (instrukcijska arhitektura), ABI (binarno sučelje aplikacije) i API.
18. **Što je API?**
	* Sučelje koje omogućava aplikacijama pristup OS funkcijama i resursima.
19. **Što je arhitektura mikrokernela?**
	* Dizajn OS-a koji odjeljuje osnovne funkcije jezgre od ostalih komponenti.
20. **Koje su prednosti mikrokernela?**
	* Fleksibilnost, jednostavnija implementacija i sigurnost u distribuiranim sustavima.
21. **Što je višedretvenost?**
	* Tehnika podjele procesa na manje niti koje se izvršavaju istovremeno.
22. **Koji su elementi višedretvenosti?**
	* Dretve koje dijele resurse unutar jednog procesa.
23. **Što je simetrična višeprocesorska obrada (SMP)?**
	* Arhitektura u kojoj OS raspoređuje procese i dretve na sve dostupne procesore.
24. **Kako SMP koristi paralelizam?**
	* SMP omogućava da više procesa ili dretvi radi istovremeno na različitim procesorima.
25. **Koje su prednosti SMP-a?**
	* Povećanje brzine, dostupnosti i skalabilnosti sustava.
26. **Što je distribuirani operativni sustav?**
	* OS koji pruža iluziju jedinstvenog memorijskog prostora i resursa na više računala.
27. **Što je objektno orijentirani dizajn OS-a?**
	* Dizajn koji koristi modularne komponente za proširenje funkcionalnosti OS-a.
28. **Kako OS osigurava sigurnost i zaštitu podataka?**
	* Kroz kontrolu pristupa, enkripciju i praćenje aktivnosti korisnika.
29. **Koji su glavni problemi u sigurnosti informacija?**
	* Povjerljivost, integritet, dostupnost i autentičnost podataka.
30. **Što je tolerancija na greške?**
	* Sposobnost OS-a da nastavi rad unatoč prisutnosti grešaka u hardveru ili softveru.
31. **Koje tehnike pomažu u toleranciji na greške?**
	* Redundancija, otkrivanje i ispravljanje pogrešaka.
32. **Što je redundancija u OS-u?**
	* Dodavanje dodatnih komponenti koje omogućuju rad OS-a u slučaju kvara.
33. **Kako OS upravlja procesima?**
	* Kreira, praćenje, planira i dodjeljuje resurse procesima.
34. **Što je proces?**
	* Jedinica aktivnosti u OS-u koja uključuje kod, podatke i kontekst izvršavanja.
35. **Koja su glavna stanja procesa?**
	* Novi, spreman, u izvođenju, blokiran i izlaz.
36. **Što je Process Control Block (PCB)?**
	* Struktura podataka koja sadrži informacije o stanju i resursima procesa.
37. **Kako OS koristi PCB?**
	* Praćenjem stanja procesa i upravljanjem prebacivanja među procesima.
38. **Što je sustav za upravljanje datotekama?**
	* OS komponente koje osiguravaju organizaciju, pohranu i pristup podacima na sekundarnoj memoriji.
39. **Kako OS organizira datoteke?**
	* Pomoću strukture direktorija, pristupnih prava i hijerarhijskog organiziranja.
40. **Što je sigurnosni referentni monitor?**
	* Komponenta koja provjerava pristup podacima i kontrolira autorizaciju korisnika.
41. **Što je virtualna memorija?**
	* Tehnologija koja koristi sekundarnu memoriju kao proširenje glavne memorije.
42. **Kako funkcionira virtualna memorija?**
	* Kroz preslikavanje virtualnih u fizičke adrese i tehnike poput straničenja i segmentacije.
43. **Što je segmentacija?**
	* Metoda upravljanja memorijom gdje se procesi dijele na logičke blokove varijabilne veličine.
44. **Koji su ciljevi segmentacije?**
	* Poboljšati učinkovitost pristupa podacima i organizirati memorijske resurse.
45. **Što je upravljanje napajanjem?**
	* Sustav za kontrolu potrošnje energije uređaja kako bi se smanjila potrošnja.
46. **Kako funkcioniraju Wakelockovi u Androidu?**
	* Omogućuju aplikacijama da zadrže sustav aktivnim, čak i kada je u stanju mirovanja.
47. **Koje su prednosti otvorenog koda u Android OS-u?**
	* Fleksibilnost, transparentnost i prilagodljivost prema potrebama zajednice.
48. **Što je Application Framework u Androidu?**
	* Sloj u Androidu koji pruža API-je za pristup ključnim funkcijama sustava.
49. **Koje su ključne komponente Application Frameworka?**
	* Upravljanje aktivnostima, prozorima, paketima i resursima.
50. **Što je modularna struktura Linux OS-a?**
	* Organizacija OS-a kao zbirke modula koji se dinamički povezuju i razdvajaju prema potrebi.

**50 Pitanja i odgovora za prezentaciju "Procesi"**

1. **Što je proces?**
	* Proces je instanca programa u izvođenju, koji uključuje kod programa, podatke i stanje izvršenja.
2. **Koja su dva osnovna elementa procesa?**
	* Programski kod i skup podataka povezanih s tim kodom.
3. **Što je identifikator procesa?**
	* Jedinstveni broj koji operacijski sustav dodjeljuje svakom procesu za praćenje.
4. **Što je Process Control Block (PCB)?**
	* PCB je struktura podataka koja sadrži ključne informacije o procesu, uključujući identifikator, stanje i memorijske pokazivače.
5. **Koje informacije sadrži PCB?**
	* Identifikator procesa, stanje, prioritet, programski brojač, memorijske pokazivače, kontekst i informacije o I/O statusu.
6. **Što je stanje procesa?**
	* Stanje u kojem se proces trenutno nalazi, kao što su novo, spreman, blokiran, izvršavanje ili izlaz.
7. **Koja su osnovna stanja u modelu procesa?**
	* Novo, spreman, u izvršenju, blokiran i izlaz.
8. **Što je dispečer?**
	* Mali program koji prebacuje procesor između različitih procesa.
9. **Koji su razlozi za stvaranje novog procesa?**
	* Novi posao, interaktivni pristup, usluge OS-a ili inicijativa korisničkog programa.
10. **Kako OS kreira novi proces?**
	* Na zahtjev drugog procesa, dodjeljuje jedinstveni identifikator i inicijalizira PCB.
11. **Što je proces roditelj?**
	* Proces koji stvara novi proces, poznat kao dijete.
12. **Koji su razlozi za gašenje procesa?**
	* Normalan završetak, prekoračeno vremensko ograničenje, nedostatak memorije, greška u zaštiti, aritmetička pogreška, korisnički zahtjev ili gašenje od strane roditelja.
13. **Što je normalan završetak procesa?**
	* Kada proces sam označi završetak svog rada pozivom OS-a.
14. **Što je model s pet stanja procesa?**
	* Model koji uključuje stanja: novo, spreman, blokiran, izvršenje i izlaz.
15. **Kako OS upravlja prebacivanjem između procesa?**
	* Spremanjem konteksta trenutnog procesa i učitavanjem konteksta drugog procesa.
16. **Što je blokiran proces?**
	* Proces koji čeka na događaj, poput završetka I/O operacije.
17. **Što znači da je proces suspendiran?**
	* Proces je privremeno premješten iz glavne memorije na disk radi oslobađanja resursa.
18. **Koji su razlozi za suspenziju procesa?**
	* Zamjena zbog nedostatka memorije, zahtjev OS-a ili korisnika, vremenski razlog ili zahtjev roditelja.
19. **Koja je uloga tablice procesa?**
	* Tablica sadrži podatke o stanju i resursima svakog aktivnog procesa.
20. **Što su informacije o kontroli procesa?**
	* Podaci potrebni OS-u za praćenje i raspoređivanje procesa, uključujući stanje, prioritete i međusobne veze procesa.
21. **Što je međuprocesna komunikacija (IPC)?**
	* Mehanizmi koji omogućavaju komunikaciju i sinkronizaciju među procesima.
22. **Koje vrste registara koriste procesori u korisničkom načinu rada?**
	* Korisnički vidljivi registri, kontrolni i statusni registri te pokazivači na stog.
23. **Što su kontrolne tablice u OS-u?**
	* Strukture podataka koje OS koristi za praćenje procesa, memorije, I/O uređaja i datoteka.
24. **Što je model procesa s dva stanja?**
	* Model u kojem proces može biti u stanju izvršavanja ili neizvršavanja.
25. **Kako OS koristi program counter?**
	* Brojač programa označava adresu sljedeće instrukcije za izvođenje.
26. **Što je kontekst procesa?**
	* Podaci poput registara i brojača programa, koji su potrebni za nastavak izvršavanja procesa nakon prekida.
27. **Koji su koraci prilikom prebacivanja procesa?**
	* Spremanje trenutnog konteksta, ažuriranje PCB-a i učitavanje novog konteksta.
28. **Što znači višezadaćno okruženje?**
	* OS omogućuje izvršenje više procesa istovremeno dijeljenjem procesorskog vremena.
29. **Koje su prednosti multiprogramiranja?**
	* Povećana efikasnost procesora i smanjenje vremena čekanja.
30. **Kako proces komunicira s OS-om?**
	* Kroz sistemske pozive.
31. **Što je sigurnosni način rada procesa?**
	* Procesi se izvode s ograničenim pravima za zaštitu sustava.
32. **Što je procesni trag?**
	* Slijed instruktivnih operacija koje proces izvršava.
33. **Kako se modeliraju veze među procesima?**
	* Kroz strukture kao što su redovi čekanja, prsteni ili druge hijerarhijske organizacije.
34. **Koji su tipični problemi kod međuprocesne sinkronizacije?**
	* Nepravilna sinkronizacija, međusobno isključivanje i zastoji.
35. **Kako OS dodjeljuje resurse procesima?**
	* Prema zahtjevu i prioritetima procesa, koristeći tablice za memoriju, I/O i datoteke.
36. **Što je procesna izolacija?**
	* Osigurava da procesi nemaju pristup podacima drugih procesa.
37. **Što je resursni zastoj (deadlock)?**
	* Situacija u kojoj dva ili više procesa čekaju na resurse koje drže drugi procesi, što onemogućuje napredak.
38. **Kako OS upravlja redovima čekanja?**
	* OS koristi redove za spremne, blokirane i suspendirane procese kako bi kontrolirao redoslijed izvršavanja.
39. **Što su privilegije procesa?**
	* Dopuštenja dodijeljena procesu, koja određuju koje resurse i instrukcije može koristiti.
40. **Kako OS reagira na greške u procesu?**
	* OS može zaustaviti, ponovo pokrenuti ili završiti proces ovisno o prirodi greške.
41. **Kako funkcionira prekinuto izvršavanje procesa?**
	* Procesor prekida trenutni proces, sprema kontekst, i izvršava program za rukovanje prekidima.
42. **Koje informacije sadrži korisnički stog procesa?**
	* Parametre funkcija i adrese povratka za procedure i pozive sustava.
43. **Kako OS koristi sustavne pozive za kontrolu procesa?**
	* Sustavni pozivi omogućavaju kreiranje, upravljanje i završetak procesa.
44. **Kako OS organizira procese u memoriji?**
	* OS koristi strukture poput tablica segmenata ili stranica za organizaciju virtualne memorije.
45. **Što je procesna slika?**
	* Kombinacija programskog koda, podataka, stoga i atributa koja opisuje proces.
46. **Kako OS dodjeljuje ID brojeve procesima?**
	* OS automatski dodjeljuje jedinstvene identifikatore za svaki novostvoreni proces.
47. **Što su korisnički vidljivi registri?**
	* Registri koji se mogu direktno referencirati iz korisničkih programa.
48. **Što su kontrolni i statusni registri?**
	* Registri koji određuju stanje procesora i omogućuju OS-u kontrolu nad izvršavanjem.
49. **Koja je svrha pokazivača na stog?**
	* Svaki proces koristi stog za praćenje procedura i prijenosa podataka.
50. **Kako OS koristi tablice za međusobnu komunikaciju procesa?**
	* Tablice sadrže oznake, signale i poruke za komunikaciju među procesima, osiguravajući pravilno djelovanje.

Ovo su pitanja i odgovori za prezentaciju o procesima. Prelazim na sljedeći dokument.

**50 Pitanja i odgovora za prezentaciju "Dretve"**

1. **Što je dretva?**
	* Dretva je jedinica izvršenja unutar procesa koja omogućuje paralelno izvođenje više dijelova istog procesa.
2. **Koje su osnovne prednosti korištenja dretvi?**
	* Manje vremena za stvaranje nove dretve, manje vremena za prebacivanje između dretvi i povećana učinkovitost u komunikaciji između dijelova istog procesa.
3. **Što je multithreading?**
	* Sposobnost operacijskog sustava da podrži više istovremenih puteva izvršenja unutar jednog procesa.
4. **Koja je razlika između jednostrukog i višestrukog pristupa dretvama?**
	* Jednostruki pristup znači jednu dretvu po procesu, dok višestruki pristup omogućava više dretvi po procesu.
5. **Što je primjer OS-a s jednom dretvom po procesu?**
	* MS-DOS je primjer jednostrukog pristupa dretvama.
6. **Koji je primjer OS-a s višestrukim pristupom dretvama?**
	* Java runtime okruženje, koje podržava više dretvi unutar jednog procesa.
7. **Što je kontrolni blok dretve (TCB)?**
	* Struktura koja sadrži informacije o trenutnom stanju dretve, uključujući programski brojač i registre.
8. **Što znači višedretvenost u OS-u?**
	* Omogućava da proces sadrži više dretvi koje se mogu paralelno izvršavati.
9. **Koja su osnovna stanja dretvi?**
	* Pokrenuto, spremno, blokirano, završeno.
10. **Što je spawn operacija kod dretvi?**
	* Stvaranje nove dretve unutar postojećeg procesa.
11. **Što se događa kad se dretva blokira?**
	* Dretva se premješta u stanje blokiranja i čeka događaj poput završetka I/O operacije.
12. **Što znači suspendiranje procesa?**
	* Kad se proces suspendira, sve njegove dretve također su suspendirane.
13. **Kako završetak procesa utječe na dretve?**
	* Završetak procesa uzrokuje prekidanje svih dretvi unutar tog procesa.
14. **Koje su ključne operacije dretvi?**
	* Spawn, blokiranje, deblokiranje i završetak.
15. **Što je korisnička dretva (ULT)?**
	* Dretva kojom upravlja aplikacija, a ne OS kernel.
16. **Što je jezgrina dretva (KLT)?**
	* Dretva kojom upravlja OS kernel i koja omogućuje paralelno izvršavanje.
17. **Koje su prednosti korisničkih dretvi?**
	* Brže prebacivanje, fleksibilan raspored specifičan za aplikaciju i kompatibilnost s raznim OS-ima.
18. **Koji su nedostaci korisničkih dretvi?**
	* Blokiranje jedne dretve blokira sve dretve u procesu i nemogućnost iskorištavanja višeprocesiranja.
19. **Kako korisničke dretve prevladavaju blokiranje?**
	* Tehnika „jacketing“ omogućava pretvaranje blokirajućih sistemskih poziva u neblokirajuće pozive.
20. **Koje su prednosti jezgrinih dretvi?**
	* Kernel može istovremeno izvršavati više dretvi iz istog procesa i omogućuje paralelno izvršavanje na više procesora.
21. **Koji su nedostaci jezgrinih dretvi?**
	* Prebacivanje između dretvi zahtijeva kernel način rada, što je sporije od korisničkih dretvi.
22. **Što je kombinirani pristup?**
	* Kombinira korisničke i jezgrine dretve, omogućavajući većinu upravljanja dretvama u korisničkom prostoru.
23. **Koji OS koristi kombinirani pristup?**
	* Solaris koristi kombinirani pristup s korisničkim i jezgrinim dretvama.
24. **Što je 1:1 odnos dretvi i procesa?**
	* Svaka dretva izvršenja je jedinstveni proces s vlastitim adresnim prostorom, primjer je tradicionalni UNIX.
25. **Što je M:1 odnos dretvi i procesa?**
	* Više dretvi unutar jednog procesa koji dijeli resurse, poput Windows NT-a i Linuxa.
26. **Što je M**

**model dretvi?**

* + Kombinacija M:1 i 1

modela, koristi se u sustavima poput Solaris-a.

1. **Koje su aplikacije pogodne za višedretvenost?**
	* Višedretvene aplikacije, Java aplikacije, aplikacije s više instanci.
2. **Kako višedretvenost utječe na pozadinske zadatke?**
	* Omogućuje izvršavanje pozadinskih zadataka bez smetnji korisničkom iskustvu.
3. **Što je hijerarhija prioriteta procesa u Windowsu?**
	* Koristi se za određivanje koji procesi će se završiti pri povratu resursa, počevši s najnižim prioritetom.
4. **Kako Android koristi dretve u aplikacijama?**
	* Svaka aplikacija koristi vlastite dretve za izvođenje različitih aktivnosti i usluga.
5. **Što je aktivnost u Androidu?**
	* Komponenta aplikacije koja omogućuje korisničku interakciju.
6. **Koje su četiri vrste Android komponenti aplikacija?**
	* Aktivnosti, usluge, pružatelji sadržaja i prijemnici za emitiranje.
7. **Što je Grand Central Dispatch (GCD) u OS X-u?**
	* Mehanizam za raspoređivanje zadataka na temelju dostupnosti jezgri.
8. **Što je blok u GCD-u?**
	* Jedinica rada koja se raspoređuje u redove za izvršavanje u FIFO načinu.
9. **Koji su ključni izazovi sinkronizacije dretvi?**
	* Osigurati da dretve koje dijele resurse ne ometaju jedna drugu.
10. **Što je stog korisničkog prostora za dretve?**
	* Sadrži parametre funkcija i adrese povratka za svaku dretvu.
11. **Kako UMS upravlja dretvama u Windowsu?**
	* UMS omogućava aplikacijama da same zakazuju dretve.
12. **Što je LWP u Solarisu?**
	* Laki proces koji omogućava korisničkoj dretvi da koristi jezgrinu dretvu.
13. **Koje informacije sadrži struktura podataka LWP-a?**
	* Identifikator, prioritet, signalne maske, stog kernela i resursi.
14. **Kako Linux tretira dretve?**
	* Linux tretira dretve kao procese, koristeći clone() za dijeljenje resursa.
15. **Što je imenski prostor u Linuxu?**
	* Omogućava procesu da ima prilagođen pogled na sustav, izoliran od drugih procesa.
16. **Koje vrste imenskih prostora Linux podržava?**
	* mnt, pid, net, ipc, uts i user.
17. **Kako Android koristi višejezgrenu tehnologiju?**
	* Android koristi više dretvi za paralelno izvođenje komponenti aplikacija.
18. **Što je dedicated process u Androidu?**
	* Svaka aplikacija koristi svoj proces i virtualni stroj radi izolacije.
19. **Koja je uloga stanja aplikacija u Metro sučelju Windowsa?**
	* Aplikacije prelaze u obustavljeno stanje kad nisu u fokusu, radi uštede resursa.
20. **Što su radničke dretve u Windowsu?**
	* Skup dretvi koje izvršavaju asinkrone zadatke za aplikaciju.
21. **Što se događa s aplikacijom u pozadini u Androidu?**
	* Kada nije u fokusu, prelazi u obustavljeno stanje i čuva resurse.
22. **Koja je funkcija Live Tiles u Windowsu?**
	* Prikazuju dinamički sadržaj aplikacije bez trošenja resursa sustava.
23. **Kako OS može ukloniti pozadinske aplikacije?**
	* OS može automatski prekinuti aplikacije koje troše resurse bez obavijesti korisnika.
24. **Što znači da Android aplikacija koristi više instanci?**
	* Omogućava paralelno izvođenje različitih instanci aplikacije na više dretvi.

Ovo su detaljna pitanja i odgovori za dokument o dretvama. Sljedeći na redu je dokument prema vašim daljnjim uputama.

**50 Pitanja i odgovora za prezentaciju "Procesorsko vrijeme"**

1. **Što je dugoročno planiranje?**
	* Dugoročno planiranje odlučuje koji se procesi prihvaćaju u sustav za obradu.
2. **Što je srednjoročno planiranje?**
	* Srednjoročno planiranje upravlja stupnjem multiprogramiranja kroz zamjenu procesa između glavne memorije i diska.
3. **Što je kratkoročno planiranje?**
	* Kratkoročno planiranje, poznato kao dispečiranje, odabire koji će proces dobiti procesor za izvršenje.
4. **Koja je svrha kratkoročnog planiranja?**
	* Cilj je optimizirati vrijeme odgovora, propusnost i učinkovitost procesora.
5. **Koja su glavna stanja procesa u modelu planiranja?**
	* Novo, spremno, blokirano, izvršenje i izlaz.
6. **Što je raspoređivanje I/O operacija?**
	* Raspoređivanje koje odlučuje koji proces će koristiti I/O uređaj kad on postane dostupan.
7. **Što je cilj raspoređivanja procesora?**
	* Zadovoljiti ciljeve sustava, poput vremena odgovora i propusnosti.
8. **Koji su kriteriji za raspoređivanje orijentirani na korisnika?**
	* Vrijeme odgovora, predvidljivost i ispunjenje rokova.
9. **Koji su kriteriji za raspoređivanje orijentirani na sustav?**
	* Propusnost, iskorištenost procesora, poštenje i provođenje prioriteta.
10. **Što je vrijeme odgovora?**
	* Vrijeme od zahtjeva korisnika do početka odgovora sustava.
11. **Što znači pojam "rokovi" u raspoređivanju?**
	* Vrijeme do kojeg proces mora biti završen, važno u sustavima u stvarnom vremenu.
12. **Kako se mjere performanse raspoređivanja?**
	* Kroz kvantitativne kriterije poput vremena obrade i propusnosti.
13. **Što je propusnost?**
	* Broj dovršenih procesa po jedinici vremena.
14. **Što je iskorištenost procesora?**
	* Postotak vremena kada je procesor zauzet izvršavanjem procesa.
15. **Što je poštenje u raspoređivanju?**
	* Svi procesi trebaju biti tretirani jednako, bez izgladnjivanja.
16. **Kako se raspoređivanje prioriteta koristi u OS-u?**
	* Procesima višeg prioriteta daje se prednost u izvršavanju.
17. **Što je balansiranje resursa?**
	* Politika koja osigurava da su svi resursi sustava adekvatno korišteni.
18. **Što je First-Come-First-Served (FCFS)?**
	* Algoritam raspoređivanja koji izvršava procese prema redoslijedu dolaska.
19. **Koji su nedostaci FCFS-a?**
	* Penalizira kratke procese i može uzrokovati duga čekanja.
20. **Što je Round Robin (RR) algoritam?**
	* Svakom procesu daje se određeni vremenski kvant prije prelaska na sljedeći proces.
21. **Kako veličina vremenskog kvanta utječe na Round Robin?**
	* Prevelik kvant smanjuje reaktivnost, dok premali kvant uzrokuje česta prebacivanja.
22. **Što je Shortest Process Next (SPN)?**
	* Algoritam koji odabire proces s najkraćim očekivanim vremenom obrade.
23. **Koji je problem s SPN algoritmom?**
	* Može dovesti do izgladnjivanja dužih procesa.
24. **Što je Shortest Remaining Time (SRT)?**
	* Preventivna verzija SPN-a koja uvijek odabire proces s najkraćim preostalim vremenom obrade.
25. **Što je Highest Response Ratio Next (HRRN)?**
	* Odabire proces s najvećim omjerom čekanja u odnosu na vrijeme obrade, dajući prednost starijim procesima.
26. **Koji algoritam koristi višerazinsko povratno raspoređivanje (feedback scheduling)?**
	* Procesi se premještaju između redova različitih prioriteta na temelju vremena izvršavanja.
27. **Kako se pojam "prioritetno redanje" koristi u raspoređivanju?**
	* Algoritmi raspoređuju procese prema redovima prioriteta, dajući prednost onima s višim prioritetima.
28. **Koji su glavni tipovi odluka o raspoređivanju?**
	* Odluka kada novi proces dolazi, kada proces završi i kada se dogodi prekid.
29. **Što znači preventivno raspoređivanje?**
	* Proces može biti prekinut i zamijenjen procesom višeg prioriteta.
30. **Što znači nepreventivno raspoređivanje?**
	* Jednom kada proces započne, nastavit će se dok se ne završi ili blokira.
31. **Što je gladovanje u raspoređivanju?**
	* Situacija u kojoj proces ne dobiva procesor zbog niskog prioriteta.
32. **Što je efektivna kvantizacija vremena?**
	* Tehnika u kojoj se vrijeme dijeli na kvante za bolje iskorištavanje procesora.
33. **Koje su prednosti Round Robin algoritma?**
	* Omogućuje pošteno raspoređivanje i bolji odgovor za kratke procese.
34. **Koji je nedostatak algoritma Shortest Remaining Time?**
	* Skup je za implementaciju jer zahtijeva praćenje preostalog vremena svih procesa.
35. **Kako se koristi planiranje po fer udjelu (fair-share scheduling)?**
	* Svakom korisniku ili grupi dodjeljuje se dio procesorskog vremena na temelju njihovog udjela.
36. **Kako UNIX koristi raspoređivanje?**
	* UNIX koristi višerazinsko povratno redanje s prioritetnim redovima.
37. **Što znači pojam "nice" u UNIX raspoređivanju?**
	* Parametar koji korisnik može postaviti za smanjenje ili povećanje prioriteta procesa.
38. **Što je efekt "izgladnjivanja" procesa?**
	* Kada proces s nižim prioritetom ostaje dugo neizvršen.
39. **Koja je svrha upotrebe povratnih informacija u raspoređivanju?**
	* Omogućuje promjenu prioriteta procesa na temelju prošlog ponašanja.
40. **Što je planiranje prema vremenskom roku?**
	* Procesi se raspoređuju prema zadanim rokovima, kritično u sustavima stvarnog vremena.
41. **Kako veličina kvanta utječe na Round Robin algoritam?**
	* Veći kvant povećava efikasnost, ali smanjuje interaktivnost.
42. **Što je cilj povratnog redanja s višestrukim redovima?**
	* Učinkovito rukovanje procesima s različitim zahtjevima.
43. **Koji je glavni nedostatak algoritma FCFS?**
	* Uzrokuje duga čekanja za kratke procese kada su u redu iza dugih procesa.
44. **Što znači prioritetno planiranje s prekidima?**
	* Planer može prekinuti proces radi izvršenja procesa višeg prioriteta.
45. **Kako raspoređivanje utječe na performanse sustava?**
	* Dobro raspoređivanje smanjuje čekanja i povećava iskorištenost procesora.
46. **Koji algoritam najbolje podržava sustave stvarnog vremena?**
	* Planiranje prema vremenskom roku.
47. **Kako algoritam Feedback daje prednost procesima?**
	* Procesi koji koriste manje vremena dobivaju više prilika za izvršenje.
48. **Što je normalized response time?**
	* Omjer između stvarnog i očekivanog vremena izvršenja procesa.
49. **Koji algoritam ima minimalan dodatni utrošak?**
	* FCFS, jer ne zahtijeva složeno praćenje.
50. **Koji algoritam pruža najbolju ravnotežu između interaktivnih i batch poslova?**
	* HRRN, jer uzima u obzir i starost i duljinu procesa.

Ovo su pitanja i odgovori za prezentaciju o procesorskom vremenu. Sljedeći na redu je dokument prema vašim daljnjim uputama.

**50 Pitanja i odgovora za prezentaciju "Višeprocesorsko i višejezgreno planiranje"**

1. **Što je višeprocesorski sustav?**
	* Višeprocesorski sustav koristi više procesora za istodobno izvršavanje više zadataka.
2. **Koja su dva glavna tipa višeprocesorskih sustava?**
	* Slabo spojeni (distribuirani) i čvrsto spojeni višeprocesorski sustavi.
3. **Što je čvrsto spojeni višeprocesorski sustav?**
	* Skup procesora koji dijele zajedničku glavnu memoriju pod kontrolom OS-a.
4. **Što je slabo spojeni višeprocesorski sustav?**
	* Skup autonomnih sustava gdje svaki procesor ima svoju memoriju i I/O kanale.
5. **Što su funkcionalno specijalizirani procesori?**
	* Specijalizirani procesori koji rade pod kontrolom glavnog procesora opće namjene.
6. **Što je granularnost u kontekstu višeprocesorskog sustava?**
	* Granularnost se odnosi na veličinu jedinica obrade i razinu sinkronizacije među njima.
7. **Koje su vrste granularnosti?**
	* Fina, srednja, gruba, vrlo gruba i nezavisna paralelizacija.
8. **Što je fina granularnost?**
	* Paralelizacija unutar jednog toka instrukcija, koja zahtijeva česte sinkronizacije.
9. **Što je srednja granularnost?**
	* Paralelizacija u kojoj aplikacija koristi više dretvi unutar jednog procesa.
10. **Što je gruba granularnost?**
	* Sinkronizacija među procesima u višeprogramskom okruženju, rijetko potrebna.
11. **Što je vrlo gruba granularnost?**
	* Paralelizacija između čvorova u mreži, često u distribuiranim sustavima.
12. **Što je nezavisni paralelizam?**
	* Paralelizacija među potpuno neovisnim procesima, poput dijeljenja vremena.
13. **Što je izazov dizajna višeprocesorskih sustava?**
	* Prilagodba planiranja ovisno o granularnosti i broju procesora.
14. **Koja su tri ključna pitanja u višeprocesorskom planiranju?**
	* Dodjela procesa procesorima, multiprogramiranje na pojedinačnim procesorima i dispečiranje.
15. **Što je statičko dodjeljivanje procesa?**
	* Proces je trajno dodijeljen jednom procesoru do završetka.
16. **Što je dinamičko balansiranje opterećenja?**
	* Korištenje zajedničkog reda čekanja i dodjela procesa prvom dostupnom procesoru.
17. **Koje su dvije glavne arhitekture za dodjelu procesa?**
	* Master/Slave i Peer arhitektura.
18. **Što je Master/Slave arhitektura?**
	* Master procesor upravlja svim ključnim funkcijama, dok slave procesori izvršavaju procese.
19. **Koji su nedostaci Master/Slave arhitekture?**
	* Neuspjeh Mastera može zaustaviti cijeli sustav, a Master može postati usko grlo.
20. **Što je Peer arhitektura?**
	* Svaki procesor može samostalno odabrati proces iz reda čekanja.
21. **Koji su izazovi Peer arhitekture?**
	* OS mora osigurati da dva procesora ne odaberu isti proces i da se procesi ne izgube.
22. **Kako se organizira red čekanja u višeprocesorskim sustavima?**
	* Koristi se zajednički red čekanja za sve procesore, često s više razina prioriteta.
23. **Što je planiranje dretvi u višeprocesorskim sustavima?**
	* Upravlja se dretvama koje pripadaju jednom procesu, omogućujući istinski paralelizam.
24. **Što je dijeljenje opterećenja?**
	* Distribucija zadataka među procesorima kako bi se spriječilo mirovanje.
25. **Koji su glavni algoritmi za dijeljenje opterećenja?**
	* FCFS, najprije najmanji broj dretvi i preventivno planiranje.
26. **Koji su izazovi dijeljenja opterećenja?**
	* Izbjegavanje uskih grla zbog međusobnog isključivanja u zajedničkoj memoriji.
27. **Što je grupno planiranje?**
	* Istovremeno planiranje svih dretvi jednog procesa kako bi se smanjila sinkronizacija.
28. **Koje su prednosti grupnog planiranja?**
	* Smanjuje sinkronizacijska kašnjenja i potrebu za prebacivanjem konteksta.
29. **Što je dodjela namjenskog procesora?**
	* Svaka dretva dobiva procesor dok se aplikacija ne završi, bez prebacivanja procesa.
30. **Koji je glavni cilj dodjele namjenskog procesora?**
	* Izbjegavanje prebacivanja konteksta i povećanje učinkovitosti u velikim sustavima.
31. **Što je dinamičko planiranje?**
	* Prilagodljivo upravljanje brojem dretvi u procesu kako bi se optimiziralo opterećenje.
32. **Koji su benefiti dinamičkog planiranja?**
	* Omogućuje bolje iskorištavanje resursa u aplikacijama koje mogu prilagoditi broj dretvi.
33. **Kako funkcionira dijeljenje predmemorije u višejezgrenim sustavima?**
	* Više dretvi može dijeliti predmemoriju, što može izazvati natjecanje za resurse.
34. **Kako Linux podržava višeprocesorske sustave?**
	* Linux koristi različite strategije raspoređivanja za real-time i ne-real-time dretve.
35. **Koji su glavni algoritmi raspoređivanja u Linuxu?**
	* SCHED\_FIFO, SCHED\_RR za real-time i SCHED\_NORMAL za ne-real-time dretve.
36. **Što je Completely Fair Scheduler (CFS)?**
	* Linux planer koji osigurava pošten pristup procesora svim zadacima koristeći virtualno vrijeme.
37. **Kako CFS koristi crveno-crno stablo?**
	* CFS koristi crveno-crno stablo za brzo dodjeljivanje zadataka s najmanjim virtualnim vremenom.
38. **Koji je algoritam planiranja u UNIX-u?**
	* UNIX koristi skalu prioriteta s klasama real-time, kernel i time-shared prioriteta.
39. **Kako FreeBSD podržava SMP i višejezgrene sustave?**
	* FreeBSD planer koristi afinitet procesora kako bi optimizirao performanse u SMP sustavima.
40. **Što je afinitet procesora?**
	* Tehnika kojom se dretva dodjeljuje istom procesoru kako bi se poboljšala učinkovitost predmemorije.
41. **Koje su dvije metode migracije dretvi u FreeBSD-u?**
	* Povlačenje, gdje slobodan procesor preuzima dretvu, i guranje, koje ravnomjerno distribuira opterećenje.
42. **Kako Windows podržava višeprocesorske sustave?**
	* Windows koristi mekani i tvrdi afinitet kako bi dretve ostale na istim procesorima kad je to moguće.
43. **Što je mekani afinitet u Windowsu?**
	* Planer pokušava zadržati dretvu na procesoru gdje je već izvršavana.
44. **Što je tvrdi afinitet u Windowsu?**
	* Korisnik može ograničiti dretve na specifične procesore, što ih sprečava u migraciji na druge jezgre.
45. **Kako funkcionira varijabilna klasa prioriteta u Windowsu?**
	* Prioritet dretve može se privremeno povećavati kako bi se osigurao brzi odgovor.
46. **Koje razine prioriteta postoje u Windowsu?**
	* Dvije razine prioriteta: real-time i varijabilna.
47. **Što je prag interaktivnosti u FreeBSD-u?**
	* Definirani prag koji označava omjer vremena mirovanja i vremena rada dretve.
48. **Kako planer Linux 2.6 koristi O(1) algoritam?**
	* Planer koristi O(1) algoritam za konstantno vrijeme odabira zadatka, bez obzira na opterećenje.
49. **Što je cilj Completely Fair Scheduler-a (CFS) u Linuxu?**
	* Modelirati idealan višezadaćni procesor s poštenim pristupom svih zadataka.
50. **Koja su glavna poboljšanja planiranja u višeprocesorskim i višejezgrenim sustavima?**
	* Poboljšana raspodjela opterećenja, afinitet procesora, fleksibilnost i optimizacija resursa.

**50 Pitanja i odgovora za prezentaciju "Međusobno isključivanje i sinkronizacija"**

1. **Što je međusobno isključivanje?**
	* Tehnika koja osigurava da kada jedan proces pristupa dijeljenim resursima, nijedan drugi proces to ne može činiti istovremeno.
2. **Zašto je važna sinkronizacija u višeprocesnim sustavima?**
	* Sinkronizacija sprječava probleme poput utrkivanja i osigurava pravilno dijeljenje resursa među procesima.
3. **Koji su glavni problemi upravljanja istodobnim procesima?**
	* Utrkivanje, zastoj, izgladnjivanje i neispravno dijeljenje resursa.
4. **Što je kritični odjeljak?**
	* Dio koda unutar procesa koji pristupa zajedničkim resursima i ne smije se istovremeno izvršavati u drugim procesima.
5. **Što je utrkivanje?**
	* Problem koji se javlja kada više procesa istodobno pristupa i modificira iste podatke, uz rezultat koji ovisi o redoslijedu pristupa.
6. **Što je izgladnjivanje?**
	* Situacija u kojoj proces ne dobiva procesorsko vrijeme zbog nedostatka prioriteta ili stalne zauzetosti resursa.
7. **Što je zastoj?**
	* Situacija u kojoj procesi čekaju jedan na drugog, stvarajući ciklus ovisnosti koji onemogućava daljnje izvršavanje.
8. **Koji su osnovni uvjeti za međusobno isključivanje?**
	* Samo jedan proces smije biti u kritičnom odjeljku, blokirani procesi ne smiju ometati druge, i nema trajnog zastoja ili izgladnjivanja.
9. **Što je Dekkerov algoritam?**
	* Prvi poznati softverski algoritam za međusobno isključivanje dvaju procesa.
10. **Što je Petersonov algoritam?**
	* Softverski algoritam koji osigurava međusobno isključivanje za dva procesa koristeći varijable zastavica i redoslijed izvršenja.
11. **Koje su vrste semafora?**
	* Cjelobrojni (brojeći) semafori i binarni semafori.
12. **Kako se koristi semafor za međusobno isključivanje?**
	* Proces koristi semWait prije ulaska u kritični odjeljak i semSignal pri izlasku.
13. **Što je mutex?**
	* Slično binarnom semaforu, omogućuje zaključavanje dijeljenih resursa, a proces koji ga zaključa mora ga i otključati.
14. **Što je monitor?**
	* Konstrukcija koja upravlja pristupom procesima dijeljenim resursima koristeći postupke pristupa i varijable uvjeta.
15. **Kako monitori upravljaju sinkronizacijom?**
	* Korištenjem varijabli uvjeta koje omogućuju zaustavljanje i nastavak izvršavanja ovisno o ispunjenju uvjeta.
16. **Što je varijabla uvjeta?**
	* Varijabla unutar monitora koja omogućuje sinkronizaciju procesa u kritičnim odjeljcima.
17. **Koje su osnovne operacije na semaforima?**
	* semWait za smanjenje vrijednosti i eventualno blokiranje, te semSignal za povećanje i eventualno deblokiranje.
18. **Koji su nedostaci hardverskog pristupa međusobnom isključivanju?**
	* Visoka potrošnja procesorskog vremena i neučinkovitost u višeprocesorskim sustavima.
19. **Što je spinlock?**
	* Tehnika međusobnog isključivanja u kojoj proces kruži u petlji čekanja dok resurs ne postane dostupan.
20. **Što je usporedi i zamijeni (compare-and-swap) instrukcija?**
	* Hardverska instrukcija koja atomarno uspoređuje i ažurira vrijednost varijable, podržavajući međusobno isključivanje.
21. **Koji je problem sa spinlockovima?**
	* Procesi troše procesorsko vrijeme u petlji, što može uzrokovati izgladnjivanje.
22. **Što je problem proizvođača i potrošača?**
	* Klasičan problem sinkronizacije u kojem proizvođači dodaju podatke u međuspremnik, a potrošači ih uklanjaju.
23. **Kako binarni semafori pomažu u međusobnom isključivanju?**
	* Djeluju kao signal koji omogućava samo jednom procesu da pristupi dijeljenom resursu u svakom trenutku.
24. **Koje su razlike između jakih i slabih semafora?**
	* Jaki semafori koriste FIFO red, dok slabi nemaju određeni redoslijed deblokiranja.
25. **Kako se sinkronizacija provodi slanjem poruka?**
	* Procesi razmjenjuju poruke za sinkronizaciju ili za prijenos podataka.
26. **Što je izravno adresiranje u slanju poruka?**
	* Pošiljatelj eksplicitno specificira identifikator odredišnog procesa.
27. **Što je neizravno adresiranje?**
	* Poruke se šalju u poštanski sandučić iz kojeg ih procesi preuzimaju.
28. **Koja je prednost FIFO reda kod slanja poruka?**
	* Osigurava pravednost u redoslijedu procesuiranja poruka.
29. **Kako se blokiranje koristi u sinkronizaciji slanjem poruka?**
	* Pošiljatelj ili primatelj čeka na izvršenje dok se poruka ne preda ili ne primi.
30. **Što je problem čitatelja i pisca?**
	* Problem u kojem više čitatelja može pristupiti podacima istovremeno, ali samo jedan pisac u isto vrijeme.
31. **Koji su uvjeti za rješenje problema čitatelja i pisca?**
	* Više čitatelja može čitati istovremeno, ali samo jedan pisac može pisati dok su čitatelji blokirani.
32. **Što je livelock?**
	* Situacija u kojoj procesi nastavljaju mijenjati stanja bez korisnog rada.
33. **Kako onemogućavanje prekida osigurava međusobno isključivanje?**
	* Proces se nastavlja bez mogućnosti prekida, sprječavajući paralelne pristupe dijeljenim resursima.
34. **Koji su nedostaci onemogućavanja prekida?**
	* Smanjena učinkovitost i neprikladnost za višeprocesorske sustave.
35. **Kako slabi semafori upravljaju redoslijedom pristupa?**
	* Nemaju definirano pravilo redoslijeda deblokiranja procesa.
36. **Kako monitori podržavaju višestruki ulaz?**
	* Samo jedan proces može pristupiti monitoru, osiguravajući međusobno isključivanje.
37. **Što je raspored u poštanskim sandučićima?**
	* Pristup koji omogućava slanje poruka u zajedničke redove za pohranu do preuzimanja.
38. **Koja je razlika između send/receive funkcija u slanju poruka?**
	* send šalje poruku, dok receive prima poruku s određenog izvora ili bilo kojeg izvora.
39. **Što je csignal(c) u monitorima?**
	* Funkcija koja deblokira proces blokiran uvjetom c.
40. **Što je problem međusobnog isključivanja kod operativnih sustava?**
	* Potreba da se osigura pristup dijeljenim resursima bez međusobnog ometanja.
41. **Koja je osnovna uloga binarnog semafora?**
	* Omogućava samo jednom procesu da pristupi resursu u svakom trenutku.
42. **Koje su operacije potrebne za rješenje problema proizvođača i potrošača?**
	* semWait i semSignal za upravljanje međuspremnikom i sprječavanje istovremenog pristupa.
43. **Što je slabost softverskih algoritama za međusobno isključivanje?**
	* Složenost i ograničena primjenjivost u višeprocesorskim sustavima.
44. **Kako semafori osiguravaju sinkronizaciju?**
	* Semafori omogućuju blokiranje i deblokiranje procesa prema stanju semafora.
45. **Kako se provodi međusobno isključivanje pomoću strojnih instrukcija?**
	* Korištenjem atomskih operacija poput compare-and-swap.
46. **Što je varijabla uvjeta u monitorima?**
	* Koristi se za čekanje dok određeni uvjet ne bude ispunjen.
47. **Koji problem rješava binarni semafor?**
	* Omogućava jedinstven pristup kritičnom dijelu.
48. **Kako se implementira slanje poruka u distribuiranim sustavima?**
	* Pomoću send i receive funkcija koje omogućuju procesima komunikaciju.
49. **Kako se upravlja redom čekanja na semaforu?**
	* Procesi se dodaju u red i oslobađaju prema FIFO principu ili drugim pravilima.
50. **Koje su glavne prednosti monitora?**
	* Osiguravaju jednostavnu sinkronizaciju, jasnu kontrolu pristupa i enkapsulaciju podataka.

**50 Pitanja i odgovora za prezentaciju "Prekidi"**

1. **Što je prekid?**
	* Signal poslan od računalne komponente ili programa prema OS-u, koji privremeno zaustavlja tekuće procese za obradu prekida.
2. **Koja je uloga operacijskog sustava kod prekida?**
	* OS obrađuje prekid preko posebnog programa, rukovatelja prekida (interrupt handler).
3. **Koji su primjeri prekida?**
	* Hardverska pogreška, spremnost diska, pritisak tipke na tipkovnici, programska greška.
4. **Koje su vrste prekida?**
	* Hardverski i softverski prekidi.
5. **Što je hardverski prekid?**
	* Prekid generiran od fizičkih komponenti računala, npr. kad je disk spreman za podatke.
6. **Što je softverski prekid?**
	* Prekid pokrenut od programa, npr. TRAP instrukcija za CPU.
7. **Zašto su prekidi važni?**
	* Omogućuju multitasking, daju korisnicima kontrolu nad računalom i povećavaju efikasnost.
8. **Što je prioritet prekida?**
	* Hijerarhija koja određuje koji će se prekid prvo obraditi kad se pojavi više prekida.
9. **Kako prekid može utjecati na druge procese?**
	* Prekid može prekinuti trenutne procese ili utjecati na njihovo kašnjenje.
10. **Što je IRQ?**
	* IRQ (Interrupt Request) je zahtjev za prekidom koji privremeno zaustavlja tekući program radi pokretanja rukovatelja prekida.
11. **Kako se IRQ-ovi identificiraju?**
	* Obično po brojevima, npr. tipkovnica koristi IRQ 1.
12. **Koja je uloga rukovatelja prekida (interrupt handler)?**
	* Rukovatelj prekida obrađuje zahtjev prekida prije nastavka s prekinutim procesom.
13. **Što se događa tijekom obrade prekida?**
	* Trenutni zadatak se zaustavlja, bilježi stanje i započinje obrada prekida.
14. **Kako operacijski sustavi kao Linux i Windows identificiraju IRQ-ove?**
	* Linux koristi procinfo, a Windows msinfo32.exe ili Upravitelj uređaja.
15. **Kako se rukovatelj prekida pokreće?**
	* CPU prima signal za prekid, zaustavlja tekući zadatak i prebacuje kontrolu na rukovatelja.
16. **Kako prekidi doprinose multitaskingu?**
	* Prekidi omogućuju da se aplikacije privremeno zaustave i nastave, simulirajući multitasking.
17. **Koliko tipično traju prekidi?**
	* Većina traje između 0,1% i 2% procesorskog vremena.
18. **Kada visoka učestalost prekida može signalizirati problem?**
	* Ako prekidi traju više od 5% CPU vremena, može značiti hardversku grešku.
19. **Kako korisnički prekid (npr. pritisak tipke) utječe na OS?**
	* OS reagira na signal pritiska tipke pokretanjem odgovarajuće funkcije.
20. **Što je preusmjeravanje prekida?**
	* Proces prebacivanja izvršenja na odgovarajući rukovatelj prekida.
21. **Kako se rješavaju konflikti između prekida?**
	* Prekidi s višim prioritetom mogu prekinuti obradu nižih prekida.
22. **Što je "prekid niske razine"?**
	* Signal generiran od hardverske komponente, npr. uređaja kao što su mrežne kartice.
23. **Što je "prekid visoke razine"?**
	* Prekid pokrenut od softverskog događaja, npr. programske greške.
24. **Što znači pojam TRAP u kontekstu CPU-a?**
	* TRAP je softverski prekid koji se koristi za izvođenje privilegiranih instrukcija.
25. **Kako procesor zna kamo prebaciti izvršenje kod prekida?**
	* Koristi tablicu prekida koja sadrži adrese rukovatelja za različite tipove prekida.
26. **Što je prekid programskog koda?**
	* Prekid generiran instrukcijama programa, npr. za pristup zaštitnim funkcijama OS-a.
27. **Što je ISR (Interrupt Service Routine)?**
	* ISR je rutina za rukovanje specifičnim tipom prekida.
28. **Kako IRQ-ovi doprinose efikasnosti sustava?**
	* Omogućuju prioritizaciju i brzu reakciju na hardverske događaje bez trošenja resursa.
29. **Kako OS štiti podatke tijekom obrade prekida?**
	* OS pohranjuje stanje procesa i obnavlja ga nakon što se prekid obradi.
30. **Koji je primjer softverskog prekida?**
	* Segmentation fault, koji se javlja kada program pristupi nedozvoljenom području memorije.
31. **Kako se koristi maskiranje prekida?**
	* Maskiranjem se privremeno onemogućuje određene prekide kako bi se spriječile smetnje.
32. **Koja je razlika između maskiranih i nemaskiranih prekida?**
	* Maskirani se mogu privremeno isključiti, dok se nemaskirani prekidi ne mogu zanemariti.
33. **Što su zadaci višeg prioriteta?**
	* Zadaci koji imaju prednost nad nižim prioritetima, kao što je hitna obrada hardverskog signala.
34. **Kako operacijski sustav obrađuje IRQ za tipkovnicu?**
	* Koristi IRQ 1 za prepoznavanje pritiska tipke i pokretanje odgovarajuće funkcije.
35. **Što znači višestruki prekid?**
	* Situacija kada više prekida dolazi istovremeno, pri čemu se obrađuje prvi po prioritetu.
36. **Kako se IRQ redovi organiziraju prema prioritetima?**
	* Najvažniji prekidi se obrađuju prvi, npr. uređaji za hitni odgovor imaju veći prioritet.
37. **Što se događa kad prekid dođe dok je već jedan aktivan?**
	* Prekid s višim prioritetom može prekinuti trenutni prekid, ostavljajući ga na čekanju.
38. **Koji su koraci obrade prekida?**
	* CPU prestaje s izvršavanjem, pohranjuje stanje i pokreće odgovarajući ISR.
39. **Kako rukovatelj prekida vraća izvršenje prekinutom procesu?**
	* Po završetku obrade, ISR vraća izvršenje koristeći pohranjeno stanje procesa.
40. **Kako se rješavaju sukobi između hardverskih prekida?**
	* Korištenjem prioritetnog sustava koji obrađuje najvažnije prekide prvo.
41. **Koja je svrha vremenskog prekida (timer interrupt)?**
	* Omogućuje OS-u kontrolu nad dijeljenjem vremena između procesa.
42. **Kako prekidi doprinose reaktivnosti OS-a?**
	* Omogućuju brzi odgovor na vanjske i unutarnje događaje.
43. **Kako se prekidna tablica koristi u višeprocesorskim sustavima?**
	* Svaki procesor može imati vlastitu tablicu ili dijeliti zajedničku, ovisno o arhitekturi.
44. **Što je proces s visokim prioritetom?**
	* Proces koji može prekinuti druge procese kako bi se izvršio odmah, npr. kritična I/O operacija.
45. **Kako se kontrolira redoslijed obrade višestrukih prekida?**
	* OS koristi redove čekanja organizirane prema prioritetima.
46. **Što je primjer softverskog prekida za administrativne zadatke?**
	* CTRL+ALT+DEL u Windowsu pokreće task manager kao prekid za korisničku intervenciju.
47. **Koji su primjeri događaja koji ne zahtijevaju korisničku intervenciju?**
	* Spremnost diska za unos podataka ili mrežni zahtjev.
48. **Što se događa kada korisnik pritisne kombinaciju tipki za prekid?**
	* OS obrađuje signal i izvodi zadani zadatak, npr. otvaranje izbornika.
49. **Kako se provodi dijagnostika prekida?**
	* U Windowsu koriste se alati kao msinfo32.exe za pregled IRQ postavki.
50. **Zašto je važna kontrola IRQ postavki?**
	* Omogućuje optimizaciju i rješavanje sukoba među uređajima koji dijele isti IRQ.

**50 Pitanja i odgovora za prezentaciju "Procesi i Dretve"**

1. **Što je proces?**
	* Proces je instanca programa u izvršenju, koja koristi resurse operacijskog sustava.
2. **Kako se stvara proces?**
	* Proces nastaje kada se program učita u memoriju i OS mu dodijeli resurse potrebne za izvršenje.
3. **Koje resurse koristi proces?**
	* Proces koristi registre, brojače, stog i dinamički alociranu memoriju.
4. **Što je stog procesa?**
	* Stog je memorijski prostor koji pohranjuje podatke prema principu "zadnji koji ulazi, prvi izlazi" (LIFO).
5. **Koji su osnovni elementi procesa?**
	* Registri, brojači, stog i dodijeljena memorija.
6. **Kako OS razlikuje procese?**
	* Svakom procesu OS dodjeljuje jedinstveni identifikator ili PID.
7. **Što znači da proces ima vlastiti adresni prostor?**
	* Svaki proces ima odvojeni dio memorije kojem drugi procesi ne mogu pristupiti.
8. **Što je dretva?**
	* Dretva je jedinica izvršenja unutar procesa koja dijeli resurse s drugim dretvama istog procesa.
9. **Koje su prednosti korištenja dretvi?**
	* Manje prebacivanja konteksta, lakša komunikacija i bolja efikasnost u korištenju resursa.
10. **Što se dijeli između dretvi?**
	* Adresni prostor, globalne varijable, otvorene datoteke i resursi.
11. **Koja je razlika između procesa i dretve?**
	* Proces ima vlastiti adresni prostor, dok dretve dijele resurse unutar istog procesa.
12. **Kako se odvija prebacivanje procesorskog vremena među dretvama?**
	* Koristi se time-division multiplexing za privid simultanog izvođenja na jednom procesoru.
13. **Što je višenitnost?**
	* Sposobnost operacijskog sustava da podrži više dretvi unutar jednog procesa.
14. **Što je primjer za višenitnost?**
	* Jedna dretva čita podatke s diska, dok druga izvršava aritmetičke operacije.
15. **Kako višeprocesorski sustavi koriste dretve?**
	* Dretve se mogu paralelno izvršavati na različitim jezgrama procesora.
16. **Što je kontrolna dretva?**
	* Svaki proces ima barem jednu dretvu koja upravlja izvršenjem tog procesa.
17. **Koja je razlika između globalnih varijabli i stoga dretve?**
	* Globalne varijable dijele sve dretve unutar procesa, dok stog pripada pojedinačnoj dretvi.
18. **Kako se kreira proces u C/C++?**
	* Proces se može kreirati pomoću funkcije fork().
19. **Što znači rezultat fork() funkcije?**
	* Rezultat manji od 0 označava grešku, 0 znači uspješno kreiranje djeteta, a pozitivna vrijednost vraća PID djeteta roditelju.
20. **Koja funkcija u C/C++ kreira novu dretvu?**
	* Funkcija pthread\_create() kreira novu dretvu.
21. **Što se događa kod pthread\_detach()?**
	* Odvaja dretvu od pozivateljske dretve, omogućujući joj samostalno izvršenje.
22. **Kako se razlikuju fork() i clone()?**
	* fork() stvara novi proces, dok clone() omogućava preciznu kontrolu nad dijeljenjem resursa.
23. **Koji su primjeri višedretvenih aplikacija?**
	* Web preglednici, baze podataka, aplikacije za obradu teksta.
24. **Što znači dijeljenje adresnog prostora?**
	* Sve dretve unutar istog procesa koriste isti skup adresa u memoriji.
25. **Kako OS prebacuje kontekst među procesima?**
	* Pohranjuje stanje trenutnog procesa i učitava stanje sljedećeg.
26. **Kako korisnici percipiraju multitasking na jednom procesoru?**
	* Procesorsko vrijeme se brzo prebacuje među zadacima, stvarajući privid simultanog izvođenja.
27. **Kako funkcija pthread\_join() radi?**
	* Čeka da se specifična dretva završi prije nastavka izvršenja programa.
28. **Koja su tri načina kreiranja procesa?**
	* fork(), vfork() i clone() u C/C++.
29. **Što znači PID u kontekstu procesa?**
	* PID je jedinstveni identifikator procesa u OS-u.
30. **Što je primjer procesa u OS-u?**
	* Tekstualni editor, web preglednik ili operativni sustav.
31. **Kako Python stvara novi proces?**
	* Korištenjem funkcije os.fork().
32. **Koja funkcija u Pythonu prikazuje PID?**
	* os.getpid() vraća identifikator trenutnog procesa.
33. **Koja je razlika između vfork() i fork()?**
	* vfork() suspendira roditelja dok se dijete ne završi, dok fork() omogućuje paralelno izvršavanje.
34. **Kako dretve koriste resurse unutar jednog procesa?**
	* Dijele globalne varijable i otvorene datoteke, ali imaju vlastite stogove.
35. **Što je virtualni CPU u kontekstu procesa?**
	* Pojam koji opisuje dijeljenje fizičkog CPU-a među više dretvi.
36. **Koji su izazovi korištenja dretvi?**
	* Potreba za sinkronizacijom, mogućnost izgladnjivanja i teškoće u otkrivanju grešaka.
37. **Što je pthread\_exit()?**
	* Funkcija koja završava dretvu u C/C++.
38. **Kako OS alocira resurse za nove procese?**
	* OS dodjeljuje memoriju, registre i brojače za svaki novi proces.
39. **Što znači izolacija procesa?**
	* Osigurava da procesi nemaju pristup memoriji drugih procesa.
40. **Kako se odvija komunikacija među procesima?**
	* Korištenjem tehnika kao što su cijevi, poruke i zajednička memorija.
41. **Koje su osnovne karakteristike dretvi?**
	* Dijeljenje memorije, vlastiti stog i zajednički resursi.
42. **Koji su rizici prilikom rada s dretvama?**
	* Moglo bi doći do utrkivanja i zastoja ako nije pravilno sinkronizirano.
43. **Kako dretve povećavaju efikasnost aplikacija?**
	* Omogućuju paralelno izvršavanje zadataka unutar iste aplikacije.
44. **Što znači pthread\_equal() u C/C++?**
	* Provjerava jesu li dvije dretve iste.
45. **Što se događa kada se koristi pthread\_cancel()?**
	* Funkcija omogućuje ručno završavanje određene dretve.
46. **Kako se alociraju dretve u višeprocesorskim sustavima?**
	* Svaka dretva se može izvršavati na različitoj jezgri procesora.
47. **Koji je rezultat fork() kada se izvrši tri puta?**
	* Ukupno će se stvoriti osam instanci jer svaka fork() udvostručuje broj procesa.
48. **Koje tehnike omogućuju sinkronizaciju dretvi?**
	* Korištenje semafora, mutexa i monitora.
49. **Što znači prebacivanje konteksta?**
	* Proces pohranjivanja i uč

**50 Pitanja i odgovora za prezentaciju "Boot proces OS-a"**

1. **Što je BIOS?**
	* BIOS (Basic Input/Output System) je osnovni program na EPROM čipu koji inicijalizira računalne komponente prilikom pokretanja.
2. **Koja je uloga BIOS-a u boot procesu?**
	* BIOS obavlja POST test, prepoznaje priključene uređaje i pokreće bootloader.
3. **Što je bootloader?**
	* Bootloader je program koji učitava operativni sustav iz memorije i prenosi mu kontrolu.
4. **Koje su faze boot procesa?**
	* Uključivanje napajanja, BIOS, bootloader i učitavanje operativnog sustava.
5. **Što je MBR?**
	* Master Boot Record, početni sektor na disku koji sadrži bootloader.
6. **Koja je funkcija MBR-a?**
	* MBR učitava bootloader i omogućava pokretanje operativnog sustava s diska.
7. **Što je kernel?**
	* Kernel je jezgra OS-a koja upravlja hardverskim resursima i izvršava osnovne funkcije sustava.
8. **Koja je uloga kernela?**
	* Kernel upravlja procesima, memorijom, I/O jedinicama i sigurnošću sustava.
9. **Što su prstenovi u procesoru?**
	* Prstenovi (rings) su razine pristupa, najčešće prsten 0 za kernel i prsten 3 za korisničke aplikacije.
10. **Zašto se koristi kernel mod?**
	* Kernel mod omogućuje pristup cijeloj memoriji i svim CPU instrukcijama.
11. **Što je korisnički mod?**
	* Mod u kojem rade korisničke aplikacije, ograničen pristupom za sigurnost.
12. **Kako prstenovi štite sustav?**
	* Sprečavaju neovlašteni pristup kritičnim dijelovima OS-a i hardvera.
13. **Što je POST test?**
	* Power-On Self Test, test periferne opreme nakon pokretanja računala.
14. **Što je BCDEdit?**
	* Alat za upravljanje boot konfiguracijom u Windowsima.
15. **Što je funkcija Winload.exe?**
	* Učitava operativni sustav u memoriju nakon bootloadera.
16. **Što znači NT loader (NTLDR) u Windowsima?**
	* Program koji je odgovoran za učitavanje OS-a u starijim verzijama Windowsa.
17. **Koje su glavne funkcije BIOS-a?**
	* Prepoznavanje periferija, učitavanje bootloadera i testiranje hardvera.
18. **Što je Boot.ini?**
	* Konfiguracijska datoteka u starijim Windowsima koja sadrži postavke boot procesa.
19. **Koja je funkcija HAL-a (Hardware Abstraction Layer)?**
	* Omogućuje kernelu da upravlja različitim hardverom koristeći isti kod.
20. **Što je boot sektor?**
	* Sektor diska koji sadrži boot kod za podizanje OS-a.
21. **Što se događa u Fazi 0 pokretanja Windowsa?**
	* Inicijaliziraju se procesori i memorija, kreiraju se osnovni sustavni procesi i omogućuju prekidi.
22. **Što se događa u Fazi 1 pokretanja Windowsa?**
	* Inicijaliziraju se upravljački programi, kreira se registry, te se pokreće Session Manager (Smss).
23. **Što je Smss u Windowsima?**
	* Session Manager SubSystem, ključni proces za upravljanje sesijama u OS-u.
24. **Što je Winlogon.exe?**
	* Proces odgovoran za prijavu korisnika u Windows OS.
25. **Koja je razlika između korisničkog i jezgrinog moda rada?**
	* Jezgrin mod ima potpuni pristup hardveru, dok korisnički mod ima ograničen pristup.
26. **Koja je funkcija Csrss.exe?**
	* Client/Server Runtime SubSystem, upravlja dijelovima Windows GUI-a.
27. **Kako Win8 ubrzava boot proces?**
	* Kernel prelazi u hibernaciju umjesto potpunog gašenja.
28. **Što je Fast Boot opcija u Windowsima?**
	* Opcija koja smanjuje vrijeme podizanja OS-a dijelom stavljanjem kernela u hibernaciju.
29. **Što je hibernacija u kontekstu boot procesa?**
	* Pohrana trenutnog stanja kernela na disk kako bi se smanjilo vrijeme ponovnog pokretanja.
30. **Koja je uloga kernel moda u Linuxu?**
	* Osigurava pristup svim dijelovima memorije i kontrolu nad hardverom.
31. **Što je GRUB?**
	* GRand Unified Bootloader, popularni bootloader za Linux.
32. **Što je LILO?**
	* LInux LOader, stariji bootloader koji podržava jednostavne konfiguracije.
33. **Što je initrd u Linuxu?**
	* Initial RAM Disk, koristi se za pokretanje kernela s potrebnim modulima prije montaže root particije.
34. **Što je SystemD?**
	* Moderni init sustav u Linuxu koji upravlja pokretanjem servisa i procesa.
35. **Koja je funkcija bootloadera u Linuxu?**
	* Učitava kernel i init sustav, omogućujući daljnje pokretanje OS-a.
36. **Kako Linux upravlja procesom podizanja kernela?**
	* Kernel se inicijalizira, postavlja root datotečni sustav i pokreće init ili SystemD.
37. **Što je init u Linuxu?**
	* Prvi proces koji pokreće sve ostale procese u tradicionalnim Linux distribucijama.
38. **Što je master boot record (MBR)?**
	* Prvi sektor diska koji pokreće bootloader, važan u procesu bootanja.
39. **Što znači da Windows koristi BCDEdit?**
	* Zamjenjuje Boot.ini u modernim Windowsima za pohranu informacija o pokretanju.
40. **Što su kernel prstenovi?**
	* Sigurnosne razine u procesoru koje omogućuju razdvajanje pristupa između korisničkih i sistemskih funkcija.
41. **Što se događa kada se Windows kernel učita?**
	* Kernel pokreće ključne sistemske servise i priprema sustav za prijavu.
42. **Kako boot proces pomaže u sigurnosti OS-a?**
	* Bootloader, kernel i HAL kontroliraju pristup resursima i uređajima od samog početka.
43. **Koji su popularni Linux bootloaderi?**
	* GRUB i LILO.
44. **Što je EFI particija?**
	* Particija na modernim sustavima koja sadrži boot datoteke za OS i omogućuje EFI pokretanje.
45. **Kako Mac OSX koristi prstenove?**
	* OSX također koristi prsten 0 za kernel mod i prsten 3 za korisnički mod, s ograničenom uporabom drugih prstena.
46. **Koji je postupak pokretanja za Android?**
	* Android koristi Linux kernel i bootloader, s prilagođenim init procesima.
47. **Kako ARM procesori pristupaju prstenovima?**
	* ARM procesori obično nemaju hijerarhiju prstenova kao x86 procesori, već koriste vlastite sigurnosne modove.
48. **Što je Userinit.exe?**
	* Proces koji pokreće korisničke skripte i aplikacije nakon prijave u Windows sustav.
49. **Koja je uloga Explorer.exe?**
	* Explorer je shell proces koji pruža korisničko sučelje u Windowsima.
50. **Zašto Windows 11 ima brži boot?**
	* Optimizirano je upravljanje memorijom i CPU resursima, uz Fast Boot opciju.

**50 Pitanja i odgovora za prezentaciju "CPU Vrijeme i Raspodjela Resursa"**

1. **Što je dretva?**
	* Dretva je osnovna jedinica izvršenja unutar procesa koja može biti pokrenuta na CPU-u.
2. **Koje su vrste višezadaćnosti?**
	* Preemptive (iznuđena) i Non-Preemptive (neiznuđena).
3. **Što je iznuđena višezadaćnost?**
	* CPU može prekinuti dretvu i prebaciti se na drugu, što omogućava bolje raspoređivanje resursa.
4. **Što je neiznuđena višezadaćnost?**
	* Dretva nastavlja s izvršavanjem sve dok ne završi ili se dobrovoljno preda CPU-u.
5. **Koji su ciljevi CPU raspoređivanja?**
	* Minimiziranje vremena odaziva, maksimiziranje broja operacija i pravednost.
6. **Što je FIFO (First In, First Out) algoritam?**
	* Strategija raspoređivanja u kojoj prva dretva u redu dobiva CPU sve dok se ne završi.
7. **Koje su prednosti FIFO algoritma?**
	* Jednostavna implementacija, manji broj zamjena konteksta i izbjegavanje izgladnjivanja.
8. **Koji je nedostatak FIFO algoritma?**
	* Može uzrokovati čekanje kod CPU intenzivnih zadataka.
9. **Što je Round Robin (RR) algoritam?**
	* Algoritam koji daje svakom procesu određeni vremenski kvant za izvršenje.
10. **Kako se postavlja vremenski kvant u RR algoritmu?**
	* Kvant treba biti balansiran; premalen smanjuje učinkovitost, dok prevelik produžuje odaziv.
11. **Što znači da je red u RR algoritmu kružni?**
	* Dretve koje nisu završile tijekom kvanta vraćaju se na kraj reda.
12. **Koja je osnovna ideja SJF (Shortest Job First) algoritma?**
	* Izvršava prvo zadatke s najkraćim vremenom izvršenja.
13. **Koja je razlika između SJF i SRTF algoritma?**
	* SRTF (Shortest Remaining Time First) prekida trenutnu dretvu ako nova ima kraće preostalo vrijeme.
14. **Koja su obilježja SJF algoritma?**
	* Neiznuđen je, ali može uzrokovati izgladnjivanje dugih zadataka.
15. **Koja je prednost SRTF algoritma?**
	* Brže završava kratke zadatke, ali može uzrokovati izgladnjivanje dužih zadataka.
16. **Kako se računa prosječno vrijeme čekanja?**
	* Zbrajanjem svih čekanja dretvi i dijeljenjem s brojem dretvi.
17. **Koji algoritam ima najniže prosječno vrijeme čekanja?**
	* SJF, jer daje prednost kraćim zadacima.
18. **Što je kontekst prebacivanja?**
	* Proces spremanja stanja trenutne dretve i učitavanje stanja nove dretve.
19. **Što uzrokuje visoko učestalo prebacivanje konteksta?**
	* Veće opterećenje procesora i smanjenje efikasnosti.
20. **Kako preemptive algoritmi upravljaju višezadaćnošću?**
	* Prekidaju dretve nakon određenog vremena ili dolaska višeg prioriteta.
21. **Kako FIFO algoritam upravlja višezadaćnošću?**
	* Ne prekida dretve; izvršava ih redom dolaska.
22. **Kada se koristi RR algoritam?**
	* U interaktivnim sustavima gdje je brz odaziv važan.
23. **Koji je nedostatak SJF algoritma?**
	* Može uzrokovati izgladnjivanje dugačkih zadataka zbog stalnog dolaska kraćih.
24. **Što se događa kad je kvant u RR algoritmu premalen?**
	* Procesor gubi previše vremena na prebacivanje konteksta.
25. **Što je zadatak planiranja u CPU-u?**
	* Odabir dretve koja će sljedeća koristiti CPU.
26. **Koji algoritam koristi najmanje zamjena konteksta?**
	* FIFO, jer dretva drži CPU dok ne završi ili se ne blokira.
27. **Kako se obračunava kvant za optimalan rad u RR algoritmu?**
	* Kvant se prilagođava da se smanji prebacivanje konteksta i održi brz odaziv.
28. **Što je problem izgladnjivanja?**
	* Kada dugačke dretve čekaju predugo zbog dolaska kraćih.
29. **Kako se izbjegava izgladnjivanje u SJF?**
	* Primjenom kombiniranih algoritama ili dodjelom prioriteta.
30. **Kako CPU scheduler odabire sljedeću dretvu?**
	* Na temelju postavljenog algoritma raspoređivanja.
31. **Što je kvant vremena?**
	* Maksimalno vrijeme koje dretva može koristiti CPU prije nego je prekinuta.
32. **Koja je svrha preemptive algoritma?**
	* Osigurati pravičan pristup CPU-u i povećati odaziv.
33. **Kako se upravlja redoslijedom dretvi u SRTF?**
	* Dretva s najkraćim preostalim vremenom ima prioritet.
34. **Koji algoritam je najučinkovitiji za kratke zadatke?**
	* SJF i SRTF.
35. **Što je STRCF?**
	* Shortest Time to Completion First, varijanta SJF za najkraće vrijeme izvršenja.
36. **Kako FIFO raspoređuje zadatke koji dolaze u isto vrijeme?**
	* U redoslijedu dolaska, što može produžiti čekanje CPU-intenzivnim zadacima.
37. **Koji algoritam često koristi sklopovske prekide?**
	* Round Robin, radi prebacivanja nakon kvanta vremena.
38. **Što je prednost RR algoritma?**
	* Pravedna podjela CPU-a među dretvama i brz odaziv.
39. **Kako SJF može biti neefikasan?**
	* Zbog nemogućnosti predviđanja trajanja budućih zadataka.
40. **Što je „brza blagajna” u kontekstu algoritama?**
	* Ideja da kratki zadaci brzo prolaze kroz CPU, kao što su kratki redovi u trgovinama.
41. **Koja strategija se koristi za interaktivne aplikacije?**
	* Round Robin, zbog mogućnosti održavanja kratkog vremena odaziva.
42. **Što je stanje čekanja (waiting state)?**
	* Stanje kada dretva čeka na resurs ili CPU.
43. **Kako preemptive algoritmi poboljšavaju vrijeme odaziva?**
	* Prekidom dretvi kad istekne kvant, osiguravajući priliku drugim dretvama.
44. **Koji algoritam omogućava konstantan pristup CPU-u?**
	* FIFO, jer dretva dobiva CPU sve dok se ne završi.
45. **Što je kazaljka izvršavanja u RR algoritmu?**
	* Pokazivač koji upravlja kojim redoslijedom će se dretve izvršavati.
46. **Kako se SRTF nosi s različitim dužinama zadataka?**
	* Daje prednost kratkim zadacima, omogućujući brzo izvršenje.
47. **Zašto je RR prikladan za sustave u stvarnom vremenu?**
	* Jer osigurava da se niti jedna dretva ne zadrži predugo.
48. **Kako se izračunava vrijeme čekanja u SRTF?**
	* Računa se prema preostalom vremenu dretvi u redu.
49. **Kako FIFO smanjuje potrebu za kontekstnim prebacivanjem?**
	* Zadržavanjem dretve na CPU-u dok se ne završi.
50. **Koji algoritmi omogućuju poštenu raspodjelu CPU-a?**
	* Round Robin i FIFO, jer osiguravaju da sve dretve imaju pristup CPU-u.

**50 Pitanja i odgovora za prezentaciju "MS Registry i Upravljački Programi"**

1. **Što je Windows Registry?**
	* Registry je baza podataka koja pohranjuje konfiguracijske postavke sustava Windows.
2. **Koja je svrha Registry-ja?**
	* Pohranjuje postavke sustava, aplikacija i korisnika, omogućujući prilagodbu operativnog sustava.
3. **Kako se otvara Registry Editor?**
	* Upisivanjem regedit u Run dijalog ili Command Prompt.
4. **Koji su osnovni tipovi podataka u Registry-ju?**
	* REG\_SZ, REG\_DWORD, REG\_QWORD, REG\_BINARY, REG\_MULTI\_SZ i drugi.
5. **Što je REG\_SZ?**
	* REG\_SZ predstavlja jednostavan tekstualni string.
6. **Što je REG\_DWORD?**
	* 32-bitni broj koji se koristi za pohranu vrijednosti poput konfiguracijskih postavki.
7. **Koji su glavni ključevi u Registry-ju?**
	* HKEY\_CLASSES\_ROOT, HKEY\_CURRENT\_USER, HKEY\_LOCAL\_MACHINE, HKEY\_USERS, HKEY\_CURRENT\_CONFIG.
8. **Što sadrži HKEY\_CURRENT\_USER?**
	* Postavke specifične za trenutno prijavljenog korisnika.
9. **Što je HKEY\_LOCAL\_MACHINE?**
	* Ključ koji sadrži globalne postavke za cijelo računalo, bez obzira na korisnika.
10. **Kako promijeniti postavku u Registry-ju?**
	* Otvoriti regedit, pronaći ključ i promijeniti vrijednost ručno.
11. **Što je Process Monitor?**
	* Alat iz Sysinternals paketa za praćenje aktivnosti aplikacija i Registry-ja.
12. **Kako se koristi Process Monitor za praćenje Registry-ja?**
	* Postavljanjem filtara na određene procese i aktivnosti koje čitaju ili pišu u Registry.
13. **Koja je svrha ključa HKEY\_USERS?**
	* Sadrži profile svih korisnika učitanih u sustav.
14. **Kako Process Monitor može pomoći s Notepadom?**
	* Prati postavke kao što su font i pozicija prozora koje Notepad sprema u Registry.
15. **Što je HKEY\_CLASSES\_ROOT?**
	* Ključ koji osigurava da se određena aplikacija koristi za otvaranje određenih tipova datoteka.
16. **Što predstavlja REG\_BINARY?**
	* Binarnu vrijednost koja može sadržavati bilo kakve podatke u binarnom obliku.
17. **Kako dodati novi ključ u Registry-ju?**
	* Desnim klikom na željeno mjesto u Registry-ju i odabirom New > Key.
18. **Kako izmijeniti font u Notepadu putem Registry-ja?**
	* Pronaći ključ za Notepad postavke i promijeniti vrijednost lfFaceName na željeni font.
19. **Što je HKEY\_CURRENT\_CONFIG?**
	* Ključ koji sadrži postavke trenutne hardverske konfiguracije.
20. **Kako promjene u Registry-ju utječu na OS?**
	* Mogu odmah utjecati na OS, ali ponekad zahtijevaju ponovno pokretanje.
21. **Što su upravljački programi?**
	* Softverski moduli koji omogućuju komunikaciju između OS-a i hardvera.
22. **Koje su vrste upravljačkih programa?**
	* Hardverski, datotečni, filtarski, mrežni i protokolarni upravljački programi.
23. **Što je HAL?**
	* Hardware Abstraction Layer koji upravlja pristupom hardveru.
24. **Koji alati omogućuju pregled upravljačkih programa?**
	* msinfo32, Process Explorer i regedit.
25. **Gdje se nalaze postavke za upravljačke programe u Registry-ju?**
	* U ključevi HKLM\System\CurrentControlSet\Services.
26. **Što znači Type kod upravljačkih programa?**
	* Određuje vrstu usluge, npr. kernel driver, file system driver.
27. **Kako se koristi msinfo32 za pregled upravljačkih programa?**
	* Pokretanjem msinfo32 i navigacijom do Software Environment > System Drivers.
28. **Što je SERVICE\_KERNEL\_DRIVER?**
	* Oznaka za upravljačke programe kernel moda.
29. **Kako se koriste ključevi u Registry-ju za upravljanje uslugama?**
	* Pomoću Start, Type i drugih vrijednosti u ključu usluge.
30. **Što označava SERVICE\_WIN32\_SHARE\_PROCESS?**
	* Servis se dijeli među više procesa.
31. **Koji alat prikazuje DLL module učitane u System proces?**
	* Process Explorer.
32. **Što je Context Switch Delta?**
	* Broj promjena konteksta dretve unutar intervala prikaza.
33. **Kako pristupiti Registry ključu određene usluge?**
	* Otvoriti HKLM\System\CurrentControlSet\Services i pronaći naziv usluge.
34. **Što su mrežni usmjerivači?**
	* Upravljački programi koji omogućuju prijenos podataka između lokalnog i udaljenog datotečnog sustava.
35. **Koji alat omogućava pregled sistemskih dretvi?**
	* Process Explorer s pregledom kartice Threads u svojstvima System procesa.
36. **Što označava ključ HKEY\_USERS?**
	* Pohranjuje podatke profila za sve učitane korisnike.
37. **Kako funkcionira mapiranje sistemskih dretvi na upravljačke programe?**
	* Dretve u kernel modu koriste kod iz ntoskrnl.exe ili drugih upravljačkih programa.
38. **Kako promijeniti Registry postavke za Notepad?**
	* Pronaći odgovarajući ključ i izmijeniti željene postavke poput lfFaceName.
39. **Kako pronaći odgovarajući Registry ključ za postavke aplikacije?**
	* Korištenjem Process Monitora za praćenje aktivnosti aplikacije.
40. **Koje su prednosti korištenja Process Explorera za upravljačke programe?**
	* Omogućava detaljan pregled aktivnosti dretvi i povezanih modula.
41. **Kako Registry pohranjuje konfiguraciju za hardver?**
	* Pomoću ključeva u HKEY\_LOCAL\_MACHINE pod ključem System.
42. **Što je Srv2.sys?**
	* Upravljački program povezan s SMB protokolom za dijeljenje datoteka.
43. **Što označava SERVIS\_KERNEL\_DRIVER?**
	* Upravljački program koji radi u kernel modu.
44. **Što je SMB?**
	* Server Message Block, protokol za dijeljenje datoteka na Windows sustavu.
45. **Kako Registry sprema informacije o hardveru?**
	* Kroz HKEY\_LOCAL\_MACHINE i njegovu hijerarhiju ključeva.
46. **Koja je uloga SYS datoteka?**
	* Predstavljaju binarne datoteke upravljačkih programa u kernel modu.
47. **Kako provjeriti ime računala u CMD-u?**
	* Korištenjem naredbe hostname.
48. **Kako se filtrira proces u Process Monitoru?**
	* Podešavanjem Filter opcija prema nazivu procesa, npr. notepad.exe.
49. **Što je REG\_MULTI\_SZ?**
	* Tip podataka za pohranu višestrukih stringova unutar jedne vrijednosti.
50. **Kako REG\_DWORD LITTLE\_ENDIAN razlikuje vrijednosti?**
	* Pohranjuje 32-bitne vrijednosti u little-endian formatu.