

1. **-Mi az operációs rendszer kernel módja és felhasználói módja közti különbség?**

Az operációs rendszer védelmének érdekében kettő vagy több CPU működési mód létezik : kernel mód - Minden utasítás és regiszter elérhető – és felhasználói mód - - -
- Csak „biztonságos” utasítások és regiszterek használhatók –
Korszerű processzorokra jellemző, hogy legalább két futási módjuk van. A futási módok között privilégiumi, kiváltsági, fontossági különbségek vannak. A kevésbé privilegizált mód a normál mód, más néven felhasználói mód (normal mode, user mode). A privilegizáltabb futási mód neve a védett mód, rendszer mód (kernel mode, system mode, supervisor mode stb.). A processzornak védett módban szélesebb az utasításkészlete (azaz bizonyos gépi utasításokat csak védett módban tud hiba nélkül végrehajtani) és szélesebb a címtartománya (azaz bizonyos memória címekre csak védett módban képes hiba nélkül hivatkozni).

2. **-Milyen kommunikációs típust ismerünk a perifériákkal?**

Lekérdezéssel átvitel (I/O port folyamatos lekérdezése), megszakítás használat (Nem kérgeztetjük folyamatosan, hanem a kívánt esemény bekövetkezésekor a megadott programrész kerül végrehajtásra); DMA, közvetlen memória elérés.

3. **-Mi a "virtuális gép" operációs rendszer struktúra lényege, honnan ered ez az elv?**

A virtuális gép a rétegelt modell általánosítása: nemcsak a fizikai hardvert, hanem az operációs rendszer magját is hardvernek tekinti. A virtuális gép a rendelkezésre álló hardverrel azonos interfészt nyújt. Az operációs rendszer azt az illúziót kelti, mintha minden processzus saját processzort és saját (virtuális) operatív memóriát használna.

4. **-Mi a CHS címzés?**

Cylinder-Head-Sector, mágneslemez felépítési példa.

► **CHS címzés (Cylinder– Head– Sector)**

- Példa: 1.44 MB FD
- Sávok száma: 80 (0–79)
- Fejek(cylinder) száma: 2 (0–1)
- Szektorok száma egy sávon: 18 (1–18)
- Össz. Méret: $80 \cdot 2 \cdot 18 = 2880$ szektor * 512byte

5. **-Írja le az SSTF ütemezés lényegét és jellemzőit!**

Shortest Seek Time First, leghamarabb elérhető műveletet hajtva végre, legkisebb fejmozgást részesíti előnyben. Átlagos várakozási idő kicsi, várakozási idő szórása nagy. Átviteli sávszélesség nagy. Fennáll a kiéheztetés veszélye.

6. **-Mi az i-node tábla?**

Könyvtárszerkezet. Egy fület egy i-node ír le, 15 rekeszből áll, első 12 file blokkokra mutat. Ha kevés, akkor a 13. rekesz újabb i-node-ra mutat, ami +15 rekesz, ha ez is kevés akkor a 14. rekesz is újabb i-node-ra.

7. **-Az operációs rendszerek folyamatainak milyen állapotait, állapotátmeneteit ismerjük?**

Futó, Futásra kész, ideiglenesen leállították, arra vár, hogy az ütemező CPU időt adjon a folyamatnak. **Blokkolt**, ha logikailag nem lehet folytatni a tevékenységet, mert pl. egy másik eredményére vár.

(+ **alvó** / stop jelzés hatására átlép és „alszik” / , **megállított** / Kill–STOP PID megállítja a futást /és **zombi** / Ha egy gyermek folyamat befejeződik, de szülője nem hív wait(&st) hívást, akkor a gyermek bent marad a processztáblában, ez a zombi! Initfolyamat törli a „halott zombikat”. /)

8. -A kölcsönös kizárás Peterson féle megoldásának mi a lényege? (algorithmus)

A kritikus szekció előtt minden folyamat meghívja a belépés, majd utána kilépés fv-t.

9. -Mi a szemafor?

Dijkstra által javasolt változótípus. Egy egész változó, tilosat mutat ha értéke 0, szemafor>0 akkor szabad, beléphetünk a kritikus szakaszra. Két művelet tartozik hozzá: belépéskor szemafor értékének csökkentése(down), kilépéskor növelése(up).

10. -Mi a különbség a szemafor és a mutex között?

A szemafor szabályozza az erőforráshoz való hozzáférést (P, V) amíg a mutex zárolja az erőforrást a gyorsabb folyamat számára (szemafor értéke 0 vagy 1).

11. -Mit értünk folyamatok ütemezésén?

Egy algoritmus alapján eldönti az ütemező, hogy mikor melyik folyamat futhat.

12. -Mi a "sorsjáték ütemezés" lényege, hol használják?

A folyamatok között „sorsjegyeket” osztunk szét, az kapja a vezérlést, akinél a húzott jegy van. Arányos CPU időt könnyű biztosítani, hasznos pl video szervereknél.

13. -Mit jelent a holtpont gráfmodellje?

Holtpont feltételek modellezése irányított gráfokkal, ahol egy folyamatot körrel, erőforrást négyzettel jelölünk.. Ha az erőforrások, folyamatok irányított grájában kört találunk, az holtpontot jelent.

14. -Ismertesse a bankár algoritmust! Lényege!

A bankár algoritmus minden kérés megjelenítésekor azt nézi, hogy a kérés teljesítése biztonságos állapothoz vezet-e. Biztonságos állapot egy olyan helyzet, melyből létezik olyan kezdődő állapotsorozat, melynek eredményeként mindegyik folyamat megkapja a kívánt erőforrásokat és befejeződik. Ha az algoritmus ilyen állapothoz vezet, akkor jóváhagyja, ha nem, akkor a kérést elhalasztja.

15. -Mi a POSIX?

POSIX = Portable Operating System Interface for uniX. A POSIX valójában egy minimális rendszerhívás (API) készlet, szabvány. Szabvány ANSI C-vel azonos függvénykönyvtár. Ma gyakorlatilag minden OS POSIX kompatibilis. A Windows-nakis van POSIX felülete.

16. -Mik a(z) (1., 2., 3., 4.) generációs operációs rendszerek jellemzői?

- Első generáció, 1940-1955, kapcsolótábla, relé, vákumcső. Neumann János, Institute for Advanced Studies, Princeton. Egyedi gépek, Gépi kód, egyszerű matematikai számítások. Lyukkártyák megjelenése.
- Második generáció 1955-1965, tranzisztoros rendszerek. Megbízhatóvá váltak az elemek, Géptermekek (mainframe) kialakulása, Tervezés, gyártás,

programozás, üzemeltetés fázisának elkülönülése, Lyukkártyás, szalagos egységek, kötegelt rendszer megjelenése, Fortran nyelv, Op. Rendszer (FMS, Fortranmonitor system, IBM 7094 hármasa, 1401 beolvasó – 7094 feldolgozó-1401 megjelenítő)

- Harmadik generáció, 1965-1980, integrált áramkörök megjelenése: IBM 1401 és 7094 egybeolvadása: System/360 gépcsalád. Azonos rendszerek, felépítések, kompatibilitás megjelenése, OS/360 megjelenése, ez minden gépre jó, eredmény nagy, bonyolult op. Rendszer. Multiprogramozás, multitask megjelenése. Több feladat a memóriában egyidejűleg. Spooling, időosztás megjelenése. Nincs közvetlen on-line munka. Első időosztásos rendszer, MULTICS, PDP 7->UNIX.
- Negyedik generáció, 1980-tól napjainkig, személyi számítógépek, MS Windows, integrált áramkörök, CPU fejlődés, Hálózati, osztott rendszerek, MAC, Intel x86 család...

17. -Mi a RAID(1..5), mi a működésének a lényege? RAID-ek közti különbségek?

- Több lemezt fog össze, és egy logikai egységként látja az operációs rendszer.
- RAID0: Több lemez logikai összefűzésével egy meghajtót kapunk. A lemezkapacitások összege adja az új meghajtó kapacitását. A logikai meghajtó blokkjait széttrakja a lemezekre (striping), ezáltal egy fájl írása több lemezre kerül. Gyorsabb I/O műveletek. Nincs meghibásodás elleni védelem.
- RAID1: Két független lemezből készít egy logikai egységet. Minden adatot párhuzamosan kiír mindkét lemezre. (Tükrözés, mirror) Tárolókapacitás felére csökken. Drága megoldás. Jelentős hibatűrése képesség.
- RAID 1+0: Tükrös diszkekből vonjunk össze többet.
- RAID 0+1: Raid0 összevont lemezcsoportból vegyünk kettőt.
- RAID 2: Adatbitek mellett hibajavító biteket is tartalmaz. (ECC-ErrorCorrectionCode) Pl. 4 diszkhez 3 javító diszk
- RAID 3: Elég egy plusz „paritásdiszk”, $n+1$ diszk, Σn a kapacitás
- RAID 4: RAID0 kiegészítése paritásdiszkkal. Ma ezen megoldások nem gyakran használatosak.
- RAID5: Nincs paritásdiszk, ez el van osztva a tömb összes elemére. Adatok is elosztva kerülnek tárolásra. Intenzív CPU igény. Redundáns tárolás, 1 lemez meghibásodása nem okoz adatvesztést. 2 lemez egyidejű meghibásodása már igen. N lemez RAID 5 tömbben ($N \geq 3$), $n-1$ lemez méretű logikai meghajtót ad.
- RAID6: A RAID 5 paritásblokkhoz, hibajavító kód kerül tárolásra. (+1 diszk) Még intenzívebb CPU igény. Két diszk egyidejű kiesése sem okoz adatvesztést. Relatív drága. N diszk RAID 6-os tömbjének kapacitása, $N-2$ diszk kapacitással azonos.

18. -Mi a kölcsönös kizárás, mik a megvalósítás feltételei?

Kritikus programterület, szekció, az a rész mikor a közös erőforrást (memóriát) használjuk. A jó kölcsönös kizárásban nincs két folyamat egyszerre a kritikus

szekcióban, nincs CPU paraméterfüggőség, egyetlen kritikus szekción kívüli folyamat sem blokkolhat másik folyamatot és egy folyamat sem vár örökké, hogy a kritikus szekcióba tudjon belépni.

19. -Ismertesse a kölcsönös kizárás megvalósítását TSL utasítással!

A mai rendszerekben a processzornak van egy „TSL reg, lock” formájú utasítása (TSL – Test and Set Lock). Ez az utasítás beolvassa a „lock” memóriaszó tartalmát a „reg” regiszterbe, majd egy nem nulla (pl. az 1) értéket írja a „lock” memóriacímre.

Lényeges, hogy ezen művelet során a memóriasín zárolva van, tehát más CPU-k (vagy processzormagok) nem érhetik el ebben az időszakban a memóriát. Ez egy nagyon egyszerű megvalósítási lehetőséget ad arra, hogy készítsünk a TSL utasítás segítségével egy „belép”, illetve „kilép” eljárást, amelyek közé foglalhatjuk a kritikus szekciót.

20. -Ismertesse a folytonos tárkiosztás(lemez) stratégiáit, jellemzőit!

Egy elhelyezési stratégia, First Fit, Best Fit, Worst Fit(olyan memória szakaszba tesszük, hogy a lehető legnagyobb rész maradjon szabadon), veszteséges lemezkihasználás.

21. -Mi a randevú stratégia?

Üzenetküldés összegzését el lehet hagyni, ekkor ha send előtt van receive, a küldő blokkolódik, illetve fordítva. Pl.: Minix 3 is randevút használ, rögzített méretű üzenetekkel.

22. -Mi a monitor?

Magasabb szintű nyelvi konstrukció ami megoldja a szemaforok „könnyen elrontható” problémáját” Monitorban eljárások, adatszerkezetek lehetnek. Egy időben csak egy folyamat lehet aktív a monitoron belül. Ezt a fordítóprogram automatikusan biztosítja. Ha egy folyamat meghív egy monitor eljárást, akkor először ellenőrzi, hogy másik folyamat aktív-e? Ha igen, felfüggesztésre kerül. Ha nem beléphet, végrehajthatja a kívánt monitor eljárást.

23. -Mi a mutex?

A mutex egyik jelentése az angol **mutual exclusion** (kölcsönös kizárás) szóból ered. Programozástechnológiában Párhuzamos folyamatok használatakor előfordulhat, hogy két folyamat ugyanazt az erőforrást (resource) egyszerre akarja használni. Ekkor jellemzően felléphet Versengés. Ennek kiküszöbölésére a gyorsabb folyamat egy, az erőforráshoz tartozó mutexet zárol (ún. lock-ol).

Amíg a mutex zárolva van (ezt csak a zároló folyamat tudja feloldani - kivéve speciális eseteket), addig más folyamat nem férhet hozzá a zárolt erőforráshoz. Így az biztonságosan használható. (Például nem lenne szerencsés, ha DVD-írókat egyszerre két folyamat használná. :)

24. -Mi a különbség a monitor és a mutex között?

A mutex egy speciális bináris szemafor, ami 0 vagy 1. Tehát a gyorsabb folyamatnak lezárja az erőforrást. A monitor egy magasabb szintű nyelvi konstrukció, ami a megvalósításhoz használja a mutexet.

25. -Mi a soft real time rendszer?

Valós idejű rendszer egyik megvalósítása. Léteznek határidők, de ezek kismértékű elmulasztása tolerálható.

26. -Mit jelent a monitor "condition" típusa?

Mi van, ha egy folyamat nem tud továbbmenni a monitoron belül? Ennek megoldására használjuk az állapot változókat (condition). Rajtuk két művelet végezhető: wait, signal.

27. -Mi az arányos ütemezés lényege?

Az ütemezés során figyelembe vesszük a felhasználókat is. Olyan mint a garantált ütemezés, csak itt a felhasználókra vonatkoztatva.

28. -Milyen I/O eszközkategóriákat ismer? Mi a kivétel?

- blokkos eszközök
 - ❖ az információt adott méretű blokkban tárolja, mindegyiket saját címmel
 - ❖ az egyes blokkok írhatók és olvashatók az összes többi bloktól függetlenül
 - ❖ pl. lemez
- karakteres eszközök
 - ❖ vagy kibocsájtja vagy fogadja a karaktersorozatot
 - ❖ nincs blokk szerkezet
 - ❖ nem címezhető, nincs keresés
 - ❖ pl. nyomtató, egér,...

Kivételek:

- órák:
 - ❖ nem címezhetőek blokkonként
 - ❖ nem generálnak, nem fogadnak karaktersorozatot
 - ❖ csak jól meghatározott időintervallumonként megszakításokat hoznak létre
- memória leképezésű képernyők

29. -Mi az MBR?

A Master Boot Record (MBR) vagy más néven a *partíciós szektor* a merevlemez legelső szektorának (azaz az első lemezfelület első sávjának első szektorának) elnevezése. Csak a particionált merevlemezeknek van MBR-jük. A MBR a merevlemez legelején, az első partíció előtt található meg. Két részből áll:

- MBR nevű kis program, mely elindít egy operációs rendszert a merevlemezről; ezt a kis programot betöltő kódnak vagy rendszerindító kódnak (boot code) is nevezik; a
- partíciós táblát, mely a merevlemez-partíciók elhelyezkedési adatait tárolja.

30. -Mi a memóriakezelő feladata?

A folyamatok virtuális címterének címeit megfeleltesse fizikai címeknek.

31. -Mit értünk tevékeny várakozás alatt?

Mikor a CPU-t „üres” ciklusban járattuk a várakozás során, CPU időt pazarolja.

32. -Mit értünk folyamatok erőforrás görbéjén, mire használható?

33. -Mit értünk virtuális memóriakezelésen, mi a lényege? Mi a lapozás?

Egy program több memóriát is használhat mint amennyi rendelkezésre áll. A memóriában tárolt, de éppen nem használt blokkokat kiírja a külső tárolóra. A virtuális címtér „lapokra” van osztva. Lapozásnak hívjuk, mikor a logikai címből előállítjuk a fizikai címet.

34. -Mi a szoftveres és a hardveres megszakítás közti különbség? Van egyáltalán?

A szoftveres megszakítás kezelése azonos a hardveres megszakítás kezelésével.

35. -Mi az SSTF ütemezés lényege?

ShortestSeekTime First–SSTF, leghamarabb elérhetőt először. A legkisebb fejmozgást részesíti előnyben. Átlagos várakozási idő kicsi. A várakozási idő szórása nagy. Átviteli sávszélesség nagy. Fennáll a kiéheztetés veszélye.

36. -Mit nevezünk fájlrendszernek, mi köze van az FCFS ütemezéshez?

Egy fájlrendszer alatt a számítógépes fájlok tárolásának és rendszerezésének a módszerét értjük, ebbe beleértve a tárolt adatokhoz való hozzáférést és az adatok egyszerű megtalálását is. A sorrendi ütemezéssel olvashatunk és írhatunk a lemezre, aminek a rendszerét a fájlrendszer adja.

37. -Mi az Ext2FS, van-e MBR-je?

Fájlrendszer típus, minden merevlemeznek van MBR-je függetlenül a fájlerendszertől.

38. -Mi a TLB, mi a szerepe?

Translation Lookaside Buffer egy cache, amit a memória kezelő hardver használ, hogy gyorsítson a virtuális címfordítás sebességén.

39. -Honnan származik, és mi a lényege a virtuális gépek(szerver) használatának?

Az IBM-től származik az ötlet.

40. -Ismertesse a folytonos tárkiosztás(lemez) stratégiáit!

Egy elhelyezési stratégia, First Fit, Best Fit, Worst Fit(olyan memória szakaszba tesszük, hogy a lehető legnagyobb rész maradjon szabadon), veszteséges lemezkihasználás.

41. -Mi a szemafor, mikor nem használható?

Dijkstra által javasolt változótípus. Egy egész változó, tilosát mutat ha értéke 0, szemafor>0 akkor szabad, beléphetünk a kritikus szakaszra. Két művelet tartozik hozzá: belépéskor szemafor értékének csökkentése(down), kilépéskor növelése(up).

42. -Mi a Round-Robin ütemezés lényege?

Körben járó ütemezés, mindenkinek van időszelete, aminek a végén, vagy blokkolás esetén jön a következő folyamat.

43. -Milyen partíciónak nincs i-node táblája?

Minden nem UNIX fájlrendszernek nincs i-node táblája.

44. -Mi a RAID0+1 illetve RAID1+0 lemezek közti különbség?

- RAID 1+0: Tükrös diszkekből vonjunk össze többet.
- RAID 0+1: Raid0 összevont lemezcsoporthoz vegyünk kettőt.

45. -Mit nevezünk szegmentált memóriakezelésnek?

A szegmentálás egy memóriakezelési módszer. Célja a memória több címterre bontása. A memóriát logikai részekre ún. szegmensekre osztják, és minden résznek megvan a saját, 0-tól kezdődő címtartománya. Egy memóriacím így két részből áll, egy szegmenscímből és egy offset címből, azaz a memória kétdimenziós. Két szinten valósul meg, hardver és operációs rendszer szinten. A lapozással ellentétben ez nem marad rejtve a felhasználó (programozó) előtt.

46. -Mi a valós idejű ütemezés lényege?


Garantálni kell adott határidőre a tevékenység, válasz megadását.

47. -Milyen processzor védelmi szinteket ismer, hol használjuk ezeket?

Az operációs rendszer védelmének érdekében kettő vagy több CPU működési mód létezik : **kernel mód** - Minden utasítás és regiszter elérhető – és **felhasználói mód** - - - -- Csak „biztonságos” utasítások és regiszterek használhatók – Korszerű processzorokra jellemző, hogy legalább két futási módjuk van. A futási módok között privilégiumi, kiváltsági, fontossági különbségek vannak. A kevésbé privilegizált mód a normál mód, más néven felhasználói mód (normal mode, user mode). A privilegizáltabb futási mód neve a védett mód, rendszer mód (kernel mode, system mode, supervisor mode stb.). A processzornak védett módban szélesebb az utasításkészlete (azaz bizonyos gépi utasításokat csak védett módban tud hiba nélkül

végrehajtani) és szélesebb a címtartománya (azaz bizonyos memória címekre csak védett módban képes hiba nélkül hivatkozni).

48. -Ismertesse a "probléma figyelmen kívül hagyása" módszert! Hol alkalmazzák?

Nem törődünk vele, nagy valószínűséggel Ő sem talál meg bennünket, ha mégis...
Nem veszünk tudomást a holtpont létezéséről. Ha tudjuk azt, hogy egy esetleges holtpont kialakulásának valószínűsége meglehetősen kicsi, és az operációs rendszer újraindításának nincsenek kritikus következményei, akkor előfordulhat, hogy érdemes ezt a „megoldást” választani, mint holtpont  „elkerülő” algoritmust. A Unix, Windows világ is ezt a módszert használja.

49. -Mi az MFT?

Master File Table. 16 attribútum ad egy fájl bejegyzést. Minden attribútum max. 1kb. Ha ez nem elég akkor egy attribútum mutat a folytatásra. Az adat is egyfajta attribútum, így egy bejegyzés több adatsort tartalmazhat. (PL: Betekintő kép). Elvi fájl méret 2^{64} bájt lehet. Ha a fájl < 1kb, belefér az attribútumba, közvetlen fájl. Nincs fájl méret maximum.

50. -Mi a probléma a kölcsönös kizárás szigorú váltogatásos megvalósítással?

Kölcsönös kizárás feltételét teljesíti a kritikus területen lévő folyamat nem blokkolható másik folyamatot kivételével. A kölcsönös kizárás feltételeit teljesíti a 3 kivételével, ugyanis ha pl 1 folyamat a lassú, nem kritikus szekcióban van, és a 0 folyamat gyorsan belép a kritikus szekcióba, majd befejezi a nem kritikus szekciót is, akkor ez a folyamat blokkolódik mert a következő=1 lesz! (Saját magát blokkolja!)

51. -Mit jelent az "interleave" fogalma?

Lemezek forgási sebessége miatt a blokkok nem feltétlenül szomszédosak.

52. -Mit nevezünk valós idejű operációs rendszernek?

Minden egyes rendszerhívás egy előre meghatározott időn belül garantáltan végrehajtásra kerül, a konkrét körülményektől függetlenül.

53. -Milyen fájlrendszer specifikus fájlokat ismer? Hol találhatók általában?

Karakter, blokk fájlok, /dev könyvtár

54. -Mik a program-folyamat-szál közti különbségek?

Folyamat: önálló programegység, utasításszámlálóval, veremmel stb. Általában nem független folyamatok. Három állapotban lehet: futó, futásra kész, vagy blokkolt.
Szál: egy folyamaton belül több egymástól „független” végrehajtási sor. A folyamatnak önálló címtartománya van, szálnak viszont nincs.

55. -Mit nevezünk fájlrendszernek, mi köze van az FCFS ütemezéshez?

Fájlrendszer: módszer a fizikai lemezünkön, kötetünkön a fájlok és könyvtárak elhelyezés rendszerének kialakítására.

FCFS: Egy folyamat addig fut, amíg nem végez vagy nem blokkolódik. Egy pártatlan, egyszerű láncolt listában tartjuk a folyamatokat, ha egy folyamat blokkolódik, akkor a sor végére kerül.

56. -I/O szoftver modellje, milyen eszközkategóriákat ismer?

Réteges szerkezet (tipikusan 4 réteg) Hardver eszköz:

1. megszakítást kezelő réteg – legalsó kernel szinten kezelt szemafor blokkolással védve
2. eszközmeghajtó programok
3. eszköz független operációs rendszer program
4. felhasználói I/O eszközt használó program

57. -Mit nevezünk kritikus tevékenységnek?

Kritikus programterület, szekció, az a rész mikor a közös erőforrást (memóriát) használjuk.

58. -Honnan származik az op.rendszer virtualizáció, mik a jellemzői, mi köze a virtuális memóriakezeléshez?

59. -Mire szolgálnak a lapozási algoritmusok?

A lapkeret száma talán a legfontosabb információ, a lapozási algoritmus célja ennek előállítása, hiszen itt helyezkedik el az a cím, melyet egy adott hivatkozás használ.

60. -Mi a különbség folyamatok és szálak között? Van egyáltalán?

Folyamatnak önálló címtartománya van, szálnak nincs! Továbbá csak a folyamatnak van: Globális változók, Megnyitott fájl leírók, Gyermekek folyamatok, Szignálkezelők, ébresztők.

61. -Ismertesse a laptáblák szerepét! Van köze a TLB-hez?

A virtuális címet fizikai címre a laptábla segítségével lehet fordítani; de ez lassú, plusz egy memória-hozzáférést jelent. Ezért a lapkezdőcímek egy részét egy asszociatív cache-ben eltárolják, ez a TLB. Címfordításkor párhuzamosan indul a keresés a laptáblában és a TLB-ben, ha az egyikben megtalálta, akkor kész.

62. -Miért hasznos a kölcsönös kizárás üzenetküldéses megvalósítása?

Ha küldő-fogadó nem azonos gépen van, szükséges ún. nyugtázó üzenet. Ha küldő nem kapja meg a nyugtát, ismét elküldi az üzenetet. Ha a nyugta veszik el, a küldő újra küld. Ismételt üzenetek megkülönböztetése, sorszám segítségével.

63. -Mire használható a monitor?

Kölcsönös kizárás megvalósítására használjuk és ez a magasabb szintű konstrukció, sokkal biztonságosabb, mint a szemaforos megoldás.

64. -Mire jó a "Dirty-bit", hol használják?

Módosítás nyilvántartása (modified/dirty bit): minden memórialapozóhoz tartozik egy HW által kezelt bit (pl. a laptáblában) - betöltéskor törlik, módosításkor beállítják. Ha a dirty bit 1, akkor erre a lapra történt írási művelet, mióta a fizikai memóriába került. Ezt a bitet a CPU tartja karban (automatikusan), ezzel az operációs rendszer munkáját

segítve. 0 érték esetén a lapokat nem kell a diszkre kiírni, ha kiszorulnak a fizikai memóriából (hiszen a tartalmuk a fizikai memóriába helyezés óta változatlan, a diszken tárolt változat tehát továbbra is aktuális).

65. -Mi a CHS-LBA címzés közti különbség? Van egyáltalán?

A cylinder, fej, szektor szám hivatkozás helyett minden szektornak egy egyedi számot ad 0-tól (N-1)-ig, ahol N a szektorok száma a lemezen.

66. -Mi a Round-Robin címzés lényege?