Operációs rendszerek vizsga (VIMIA219) 2010. 06. 01.
Mérnök informatikus szak Teljes rendelkezésre álló idő: 60 perc

| Név: | Neptun kód: |
|---|---|

**I. Teszt kérdések** **Σ** **/ 30 pont**

**Útmutató:**

Jelölje egyértelműen a helyes választ! Karikázza be az I-t, ha az állítás igaz. Karikázza be a H-t, ha az állítás hamis. Karikázza be a ?-et, ha nem tudja a választ. Ha javítani akar a válaszon, akkor húzza át mind a három mezőt, és írja a sor végére a választ (Igaz/Hamis/Nem tudom).

Pontozás: Helyes válasz 1 pont, rossz válasz -1 pont. Kérdéscsoportonként a minimum pont 0 pont.

**1. kérdéscsoport.** Folyamatok és szálak.

| | | | | | |
|---|---|---|---|---|---|
| 1. | Egy folyamat kontextusában futó szálak csak közös memórián keresztül kommunikálhatnak. | I | H | ? | Hamis, bármilyen más megoldást, pl. üzenettovábbítást is használhatnak, de a memórián keresztüli kommunikáció a leghatékonyabb. |
| 2. | A folyamat kontextusában futó szálaknak saját verem (stack) áll rendelkezésére, de ezt a vermet bármelyik másik a folyamat kontextusában futó szál is írhatja/olvashatja. | I | H | ? | Igaz, hiszen a verem is a folyamat címterében van. Ez nyilvánvalóan egy súlyos hiba, de nem egyszer megtörtént. |
| 3. | A modern Linux operációs rendszerek szál alapú ütemezőt használnak. | I | H | ? | Hamis, taszk alapú az ütemezés, a taszk lehet folyamat vagy szál. |
| 4. | A folyamatok operációs rendszer szintű támogatásához szükséges, hogy a processzor támogassa a védelmi szinteket, és legyen MMU-ja. Ezek hiányában csak szálak valósíthatóak meg (pl. uC/OS-II). | I | H | ? | Igaz. |

**2. kérdéscsoport.** A párhuzamos programozás során elkövetett hibákkal és az azok elkerülésével kapcsolatos kérdések.

| | | | | | |
|---|---|---|---|---|---|
| 5. | A versenyhelyzet (race condition) esetén a helyes/hibás működés függ a szekvenciális részfeladatok lefutási sorrendjétől, éppen ezért az ilyen hibák okainak megtalálása és javítása nehéz. | I | H | ? | Igaz. |
| 6. | A prioritás inverzió egy speciális versenyhelyzet, ami prioritásos ütemezőt használó valós idejű rendszerekben okozhat elsősorban problémát. | I | H | ? | Igaz. A prioritás inverzió valós idejű rendszerekben érdekes, és itt hibát is okoz, hiszen nem fut le időben a magas prioritású feladat, ami ott hiba! |
| 7. | Kiéheztetésről akkor beszélünk, amikor egy feladat nem tud futó állapotba kerülni, mivel más feladatok a teljes CPU időt elhasználják. | I | H | ? | Hamis, bármilyen más közös erőforrásra is előállhat kiéheztetés. Ha a "Kiéheztetésről csak akkor |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | beszélünk" módon kezdődött volna a kérdés, akkor 100%-os lenne. Ponthatárt ezért leszállítottuk 1 ponttal. |
| 8. | Monitor alkalmazásával elkerülhetjük a versenyhelyzeteket. | I | H | ? | Hamis, a monitor csupán a lock-olás egyes alapvető hibái ellen véd, pl. lock-olás elmaradása, felszabadítás elmaradása, más erőforrás lock-olása, stb. |

**3. kérdéscsoport.** A Windows felépítésével kapcsolatos kérdések.

| | | | | | |
|---|---|---|---|---|---|
| 9. | A Windowsban az NT API nyilvános, a függvényei hivatalosan dokumentáltak. | I | H | ? | Hamis, csak a Windows API publikus, az NT API változhat a kiadások között |
| 10. | Windowsban a környezeti alrendszerek felhasználói módban futnak. | I | H | ? | Igaz. |
| 11. | Windowsban a kernel és az Executive réteg komponensei egy binárisban találhatóak. | I | H | ? | Igaz, mindkettő az ntoskrnl.exe-ben található meg. |
| 12. | A Windows az x86 architektúra által biztosított mind a négy védelmi szintet (ring) használja. | I | H | ? | Hamis, csak a ring0-t és a ring3-at |

**4. kérdéscsoport.** Windows operációs tulajdonságaival kapcsolatos kérdések.

| | | | | | |
|---|---|---|---|---|---|
| 13. | 32 bites Windows használata esetén a felhasználói folyamatok maximum 2GB virtuális címteret használhatnak. | I | H | ? | Hamis, a /3GB kapcsoló használata esetén 3 GB-ot. |
| 14. | 64 bites Windows-ok esetén a támogatott fizikai memóriát a hardver és az operációs rendszer együtt korlátozza, de az jóval kisebb, mint 2^64 byte. | I | H | ? | Igaz. |
| 15. | A Windows operációs rendszerekben a felhasználókat felhasználói névvel azonosítja az operációs rendszer. | I | H | ? | Hamis, erre a Security Identifier szolgált. Nem volt elég explicit a kérdés, az ACL-ekben azonosítják így, és nem névvel. Minden választ, és a nem tudom választ is elfogadtunk 1 ponttal. |
| 16. | A quantum az az időszelet, amíg egy szál futhat, utána újra fut az ütemező. | I | H | ? | Igaz. |

**5. kérdéscsoport.** UNIX folyamatkezeléssel kapcsolatos kérdéskör.

| | | | | | |
|---|---|---|---|---|---|
| 17. | A folyamatokat a felhasználók indítják. | I | H | ? | Hamis, mivel vannak kernel és rendszer folyamatok is. |
| 18. | Egy saját folyamatot a felhasználó jelzés (signal) segítségével leállíthat. | I | H | ? | Igaz. Lásd kill parancs. |
| 19. | A felhasználói folyamatok kernel módban is tudnak futni. | I | H | ? | Igaz, a felhasználói folyamatok kernel módban is tudnak futni, pl. amikor rendszerhívásokat hajtanak |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | végre. |
| 20. | A fork() rendszerhívás betölt egy új programkódot. | I | H | ? | Hamis, mivel csak megduplázza a futó folyamatot. |

**6. kérdéscsoport.** Mai UNIX rendszerek tulajdonságai.

| | | | | | |
|---|---|---|---|---|---|
| 21. | A Solaris csak SPARC architektúrán érhető el. | I | H | ? | Hamis, mivel x86-os rendszeren is fut. |
| 22. | A HP UNIX PA-RISC és Itanium architektúrákon fut. | I | H | ? | Igaz. |
| 23. | A ZFS snapshot készítése nagy fájlrendszer esetén hosszú ideig tart. | I | H | ? | Hamis, mivel a snapshot készítés gyors folyamat. |
| 24. | A Solaris Zóna alapvetően biztonsági feladatokat ellátó alrendszer. | I | H | ? | Hamis, mivel elsősorban virtualizációs technika. |

**7. kérdéscsoport.** Jogosultságkezelés.

| | | | | | |
|---|---|---|---|---|---|
| 25. | A jogosultságkezelő rendszerekben a hitelesítés előfeltétele az engedélyezésnek. | I | H | ? | Igen. |
| 26. | A hozzáférés vezérlési lista (access control list) többnyire egy sorrendezett lista formájában kerül megvalósításra. | I | H | ? | Igen |
| 27. | A UNIX-szerű operációs rendszer egy felhasználója átadhatja egy fájl tulajdonjogát egy másik felhasználónak. | I | H | ? | Nem, csak a rendszergazda. |

**8. kérdéscsoport.** uC/OS-II beágyazott operációs rendszer taszk állapotaival kapcsolatos kérdések.

| | | | | | |
|---|---|---|---|---|---|
| 28. | Az ISR állapotban a megszakításokat kiszolgáló taszkok kerülhetnek. | I | H | ? | Hamis, éppen a megszakítás által megszakított taszk kerül ebbe az állapotba. |
| 29. | A szunnyadó (dormant) állapotban lévő taszkokat vagy nem indították el, vagy törölték őket. | I | H | ? | Igaz. |
| 30. | Az ISR állapoton kívül bármelyik állapotban lévő taszk törölhető. | I | H | ? | Igen. A dormant állapotban lévők is, bár azokra "értelmetlen" meghívni a törlést, és az nincs is feltüntetve a kapcsolódó ábrán. A kérdés tökéletes lett volna, ha a "Az ISR állapoton kívül bármelyik állapotban lévő elindított taszk" módon kezdődött volna. |

| Név: | Neptun kód: |
|---|---|

**II/1. Nagykérdés** **Σ** **/ 10 pont**

Egy rendszerben az alábbi feladatok érkeznek be az alábbi sorrendben:

| Feladat | Beérkezési idő (ms) | CPU löket (ms) |
|---|---|---|
| P1 | 0 | 3 |
| P2 | 3 | 10 |
| P3 | 3 | 3 |
| P4 | 6 | 6 |
| P5 | 8 | 3 |

A determinisztikus modellezés módszerével vizsgálja meg, hogy a körbefordulási idő szempontjából az RR ütemezési algoritmus 4 vagy 10 ms időszelet mellet kedvezőbb-e a fenti feladatkészlet esetén? Az ütemezés időigényét hanyagolja el számításai során!
Gannt diagrammon ábrázolja a feladatok lefutását! (2-2 pont)
Számolja ki a két esetre az átlagos körbefordulási időt! (2-2 pont)
Magyarázza meg az eredményeket! Hogyan kell megválasztani az RR algoritmus esetén az időszeletet? (2 pont)

Megoldás:

4 ms időszelet:

| Időtartam | FUT | Futást kezdte | Futást befejezi | Ready sor tartalma |
|---|---|---|---|---|
| 0-3 | P1 | 0 | 3 | - |
| 3-6 | P2 | 3 | - | P3 |
| 6-7 | P2 | 3 | 7 | P3,P4 |
| 7-8 | P3 | 7 | - | P4,P2 |
| 8-10 | P3 | 7 | 10 | P4,P2,P5 |
| 10-14 | P4 | 10 | 14 | P2,P5 |
| 14-18 | P2 | 14 | 18 | P5,P4 |

| | | | | |
|---|---|---|---|---|
| 18-21 | P5 | 18 | 21 | P4,P2 |
| 21-23 | P4 | 21 | 23 | P2 |
| 23-25 | P2 | 23 | 25 | - |

Átlagos körbefordulási idő: az egyes feladatok által a rendszerbe belépéstől a kilépésig eltötltött idő átlaga, a mértékegysége s.

$t_{ta}$=((3-0)+(25-3)+(10-3)+(23-6)+(21-8))/5=(3+22+7+17+13)/5=62/5=12.4 ms

10 ms időszelet:

| Időtartam | FUT | Futást kezdte | Futást befejezi | Ready sor tartalma |
|---|---|---|---|---|
| 0-3 | P1 | 0 | 3 | - |
| 3-6 | P2 | 3 | - | P3 |
| 6-8 | P2 | 6 | 8 | P3,P4 |
| 8-13 | P2 | 8 | 13 | P3,P4,P5 |
| 13-16 | P3 | 13 | 16 | P4,P5 |
| 16-22 | P4 | 16 | 22 | P5 |
| 22-25 | P5 | 22 | 25 | - |

$t_{ta}$=((3-0)+(13-3)+(16-3)+(22-6)+(25-8))/5=(3+10+13+16+17)/5= 59/5 = 11,8 ms

A körülfordulási idő kismértékben nő a kisebb időszelet esetén, de az RR algoritmus elsősorban a válaszidő szempontjából kedvezőbb (interaktív feladatok). Az átlagos CPU löket (3+10+3+6+3)/5=25/5=5ms, ennek a 80%-ra kell beállítani az időszeletet a tapasztalati összefüggés alapján az ideális működéshez, ami pont a választott 4 ms. A 10 ms esetén az RR algoritmus lényegében átmegy a FIFO algoritmusba ennél a feladat készletnél, és ezért a P2 feltartja a többi feladat lefutását (konvoj hatás).

**II/2. Nagykérdés** **Σ / 10 pont**

Ismertesse röviden a UNIX inode felépítését. (5 pont)
Számítási példával mutassa be az adatblokk címtábla működését egy tankönyvben ismertetett fájlrendszerre! (4 pont)
Milyen statisztikára alapozva állítható be a címtábla felépítése? (1 pont)

**Megoldás**

Felépítés (1 pont / db):

- hitelesítési információk (UID, GID)
- típus (f,d,p,...)
- hozzáférési jogosultságok (tulajdonos, csoport, mindenki más)
- időbélyegek (létrehozás, módosítás, hozzáférés)
- méret
- címtábla, amely megadja az adatblokkok elhelyezkedését
- ...

A címtábla direkt és indirekt címeket tartalmaz. A direkt címek adatblokkokra mutatnak, az indirekt címek egy másik címtáblára. (Ez utóbbiban egyszeres indirekció esetén már direkt címek vannak, kétszeres indirekciónál indirekt címek.)
A fájlok mérete és méret szerinti gyakorisága közötti összefüggésre alapozva lehet beállítani a blokkméretet, valamint a címtábla felépítését. A lenti ábra szerint a fájlok kb. 80%-a kisebb, mint 12 kbájt. Pl. 12 db direkt cím, 1 bájtos cím- és 1kbájtos blokkméret esetén 12kbájt címezhető meg direkt módon, azaz a fájlok 80%-a. Ehhez 1 db egyszeresen indirekt címet felvéve a fájlméret 1MB-tal nő, ami 100% közeli eredményt hoz.