

## 2022. tavaszi előadások kvízkérdései (részlet)

A [BSc képzés tesztjei](#) linken nagyon hasonló kérdések megtalálhatóak, csak ott részletesebb az anyag, ezáltal több kérdés van.

Az ütemező átbocsájtó képessége az egységnyi időszelét alatt átütemezett taszkok száma.

- **Hamis**

A válaszidő mindig kisebb, mint a körülfordulási idő.

- **Igaz**

A körforgó (RR) ütemező kooperatív, és elkerüli a kiéheztetést.

- **Hamis**

A körforgó (RR) ütemező használata optimális átlagos várakozási időt eredményez.

- **Hamis**

A legrövidebb hátralevő löketidejű (SRTF) ütemező preemptív.

- **Igaz**

A memória-intenzív taszkok nagy memórafoglalás esetén CPU-intenzívvé válnak.

- **Hamis**

A folyamatokon belül csak egy verem lehet.

- **Hamis**

Egy általános célú operációs rendszerben jellemzőek 1-2 kontextusváltás történik másodpercenként

- **Hamis**

A szál egy szekvenciális működésű taszk, amely egy folyamaton belül más szálakkal közös halmot (heap-et) használ.

- **Igaz**

A fork() Unix rendszerhívás betölt és elindít egy új programot.

- **Hamis**

Az Apache webservert szálalapú változata nagyobb teljesítményre (kérés / mp) képes, mint a folyamat alapú.

- **Igaz**

A valós idejű működés alapvető célja az, hogy a felhasználók valós időben végezhessék a rendszeren a feladataikat.

- **Hamis**

A multiprogramozott operációs rendszer abban különbözik más rendszerektől, hogy többféle programozási nyelvet támogat.

- **Hamis**

Az időosztásos operációs rendszerek egyben multiprogramozott rendszerek is.

- **Igaz**

A Winlogon előbb fut, mint az SMSS (munkamenet-kezelő) a Windows-on.

- **Hamis**

Egy operációs rendszer nem lehet egyszerre monolitikus és moduláris felépítésű.

- **Hamis**

A modern mikrokernelek (pl. L4) nagyon lassú üzenetalapú kommunikációval működnek.

- **Hamis (?)**

Minden rendszerhívás védett módban hajtódik végre.

- **Hamis**

Egy operációs rendszer forráskódja lehet néhány tízezer programsor, de akár sok millió is.

- **Igaz**

A OS kernelek minden része (eljárása) védett módban működik.

- **Hamis**

A futási szint (runlevel) meghatározza a Unix rendszerekben futó taszkok prioritását.

- **Hamis**

A kernel moduláris felépítése csökkenti a kernel futásidejű memórafoglalását.

- **Igaz**

Az ütemező a várakozó állapotú taszkok közül választja ki a következő futó taszkot.

- **Hamis**

Az ütemező átbocsájtó képessége az egységnyi időszelét alatt átütemezett taszkok száma.

- **Hamis**

A kernel az első program, amit a háttértárról betöltve a processzor futtatni kezd.

- **Hamis**

A rendszerprogramok védett módban futnak.

- **Hamis**

Egy mikrokernel alapvetően elosztott rendszer felépítésű.

- **Igaz**

Egy párhuzamos végrehajtást (több konkurens taszk együtműködését) igénylő feladat egyetlen folyamaton belül is megvalósítható.

- **Igaz**

Ha egy memória-intenzív taszkokat futtató rendszerben alacsony a CPU-kihasználtság, akkor nincs elegendő memória a taszkok számára.

- **Igaz**

Egy rendszer absztrakt virtuális gépei összességükben több erőforrást tartalmaznak, mint amennyi fizikailag rendelkezésre áll.

- **Igaz**

A taszkok löketidejét a gyakorlatban működő ütemezők előre ismerik.

- **Hamis**

Az operációs rendszer kernelje felügyeli a felhasználói módban futó taszkok működését.

- **Hamis**

Az UNIX operációs rendszer első változata az AT&T Bell Lab kommerciális termékékeként jelent meg, amelyet számos cég és egyetem vásárolt meg.

- **Hamis**

Az első szintű (RAM, MBR) betöltő már ismeri a fájlrendszerek felépítését, hiszen onnan tölti be a második szintű betöltőt.

- **Hamis**

A rendszerhívások jellemzően megszakítással járnak együtt.

- **Igaz**

A rendszerkönyvtárak az operációs rendszer védett módban működő részei.

- **Hamis**

Egy taszk futó állapotból futásra kész állapotba kooperatív ütemező esetén is átkerülhet.

- **Igaz**

A számítógépeken futó taszkok többsége I/O-intenzív.

- **Hamis**

A(z) **átbocsájtó-képesség** meghatározza az időegység alatt elvégzett feladatok számát.

A konvoj-hatás nem jelentkezhetsz **legrövidebb hátralevő löketidejű (SRTF)** és **körforgó (RR)** ütemező esetén.

A(z) **aging/öregítés** a futásra kész taszk prioritásának növelése az ebben az állapotában eltöltött idejével arányosan.

A(z) **kiéheztetés** az a jelenség, amikor prioritásos ütemezés esetén egy taszkot folyamatosan megelőznek nála magasabb prioritásúak, ezért nem jut processzorhoz.

A(z) **FCFS** ütemező saját döntése alapján nem helyez át futásra kész állapotba taszkokat. (Ne egy konkrét ütemezőt nevezzen meg, hanem egy kategóriát!)

Állítsa párba az alábbi feladattípusokat és a rájuk leginkább jellemző feladatjellegzet!

- Önvezető autó irányítása: **valósídejű**
- Műveletek nagy adatbázisokon: **memóriaintenzív**
- Kisvállalati fájlserver: **I/O intenzív**
- Számítógépes játékok: **cpu intenzív**

```
if ((res = fork()) == 0) { // elágazás a visszatérési értékkel
    exec(...);           // gyerek ága
} else if ( res < 0 ) { // ha visszatér, exec hiba történt
    // szülő ága, hibaellenőrzés
    // ide jön a fork() hiba kezelése
}
// res = CHILD_PID (>0), szülő kódja fut tovább
```