

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 végezhessek 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. (De lehet, sőt, általában az.)

- **Hamis**

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

- **Igen**

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(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) **kooperatív** 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: **memória intenzív**
- Kisvállalati fájlszerver: **I/O intenzív**
- Számítógépes játékok: **cpu intenzív**