

## 2022. tavaszi előadások kvízkérdései (részlet)

### \tIgaz

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

A biztonsági mentés visszaállításához előbb telepítenünk kell az operációs rendszert.

Hamis

A blokkgyorsítótár a fizikai memóriát használja a diszkműveletek gyorsítására.

Igaz

A cserehely a fizikai memória egy elszeparált része, ahol a nem futó taszkok adatait tároljuk.

Hamis

A folyamatokon belül csak egy verem lehet.

Hamis

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

Hamis

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

Hamis

A keret- és a laptáblák száma megegyezik.

Hamis

A kernel adatstruktúrák egy része a fájlrendszeri interfészen keresztül hozzáférhető Linux alatt.

Igaz

A konvoj-hatás például a legrövidebb hátralevő löketidejű (SRTF) algoritmussal megszüntethető.

Igaz

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

Hamis

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

Hamis

A lapcsere során egy lap behozásakor jellemzően nem kell keretet felszabadítani, mivel szabad keretek létezéséről egy speciális folyamat gondoskodik.

Igaz

A laplopó (page daemon) valamilyen lapozási (lapbehozási) stratégiát alkalmaz.

Hamis

A leggyorsabb 1TiB méretű adattároló rendszer a merevlemez meghajtó (HDD).

Hamis

A legrégebben várakozó (FCFS) ütemezőnél akkor is jelentkezhet a konvoj-hatás, ha csak I/O intenzív taszkok vannak a rendszerben.

Hamis

A legrövidebb hátralevő löketidejű (SRTF) ütemező preemptív.

Igaz

A Linux felhasználói módban többszintű ütemezőt használ.

Igaz

A Linux kernel átütemezési pontjainak alkalmazása javítja a válaszidőt, mivel így a kernel módban futó taszkok bármikor megszakíthatók.

Hamis

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

Hamis

A memórialapú permanens tárolók (SSD-k) élettartama jellemzően 5 év körüli.

Hamis

A Microsoft RDP egy elterjedt kijelzőszerver protokoll.

Igaz

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

Igaz

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

Hamis

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

Hamis

A RAID 0 (stripe) két diszk esetén két példányban tárol minden adatot.

Hamis

A RAID0 általában gyorsabb a RAID1-nél, de a RAID1 megbízhatóbb.

Igaz

A rendszerhívások jellemzően megszakítással járnak együtt.

Igaz

A rendszerkönyvtárak olyan eljárásokat tartalmaznak, amelyek sokféle feladatban előfordulnak, így nem kell minden programban külön-külön megvalósítanunk azokat, hanem támaszkodhatunk egy közös implementációra.

Igaz

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

Igaz

A taszkok löketidejét a gyakorlatban működő ütemezők előre ismerik.

Hamis

A többszintű visszacsatolt sorok (MFQ) ütemező lefelé lépteti a taszkokat a szintek között, ha azok az adott szinten kihasználják a rendelkezésükre álló CPU-időt (pl. a RR időszeletet).

Igaz

A Unix operációs rendszer első változata az AT&T Bell Lab kommerciális termékeként jelent meg, amelyet számos cég és egyetem vásárolt meg.

Hamis

A válaszidő mindig kisebb, mint a körülfordulási idő.

Igaz

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

Hamis

A virtuális és fizikai memóriacímek futásidejű transzformációja alapvetően szoftveres úton történik.

Hamis

A Windows egyes beágyazott (kevés erőforrással rendelkező) rendszerben is működik

Igaz

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

Hamis

Az „óra” lapcsere algoritmus valójában az „újabb esély” algoritmus hatékonyabb megvalósítása.

Igaz

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

Igaz

Az időosztásos operációs rendszerek egyben multiprogramozott rendszerek is.

Igaz

Az IEEE POSIX egy szabvány, amely előírja a kernel belső felépítését.

Hamis

Az iSCSI (internet SCSI) fájlokon elvégzett műveletek hálózati átviteli protokollja.

Hamis

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

Hamis

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

Hamis

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

Igaz

Egy operációs rendszer nem lehet egyszerre monolitikus és moduláris felépítésű.

Hamis

Egy operációs rendszerben nem lehet több működő taszk, mint ahány végrehajtó egység van.

Hamis

Egy taszk futásra kész állapotból várakozó állapotba is átkerülhet.

Hamis

Ha egy felhasználói program kernel módba vált (pl. rendszerhívással), a CPU-utasításkészlete akkor is korlátozott marad, hogy ne okozzon gondot a kernelben.

Hamis

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

Igaz

Ha növeljük egy rendszerben a fizikai memória méretét, akkor mindig csökkenni fog a laphibák száma, hiszen egyszerre több lapot tarthatunk bent a memóriában.

Hamis

Két, egy folyamaton belüli szál azonos virtuális címen jellemzően ugyanazt látja, de van a virtuális címtartományuknak olyan része, amely biztosan különböző adatokat tartalmaz.

Igaz

Lehetséges várakozásmentes I/O műveletek alkalmazása a programjainkban.

Igaz

Minden rendszerhívás védett módban hajtódik végre.

Hamis

Van olyan operációs rendszer, amely részben hangutasításokkal is kezelhető.

Igaz

A legrövidebb hátralevő löketidejű ütemező (SRTF) végrehajthat **Fut** → **Futásra kész** állapotátmenetet egy taszkon, míg például a legrégebben várakozó (FCFS) nem. **Fut** → **Várakozik** állapotátmenetet egyetlen ütemező sem hajt végre taszkokon.

A **I/O-löket** a taszk működésének azon fázisa, amikor I/O művelet végrehajtására vár.

A folyamatnak saját **memóriatartománya**, a szálnak pedig saját **verme** van.

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

A konvoj-hatás nem jelentkezik **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!)

A Bélády-féle anomália azt jelenti, hogy a fizikai memóriakeretek számának **növelése** időnként a laphibák gyakoriságának **növekedését** vonja maga után.

A UNIX `mmap()`, illetve a Windows `CreateFileMapping()` rendszerhívások alapvető célja, hogy a fájlok tartalma fájlműveletek helyett **memória** műveletekkel legyen elérhető.

A Unix `flock()` egy **ajánlott** zárolási forma.

Mekkora az 5 darab 1TiB méretű diszkből összeállított RAID5 tárolórendszer teljes hasznos tárolókapacitása? **4 TiB**

A logikai kötetkezelés (LVM) egyik alapvető célja, hogy növelje a maximális **tárolórendszer-méretet**.

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

- Önvezető autó irányítása: **valósídejű**
- Műveletek nagy adatbázisokon: **memóriaintenzív**
- Kisvállalati fájlszerver: **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
```