

A 8. OLDALON VAN EGY VÁLASZTÓ, ADDIG ELŐZŐ ÉVEK ZH-I ÉS IDEI FAKULTATÍV TESZTEKBŐL VANNAK A KÉRDÉSEK ÉS VÁLASZOK.

UTÁNA INNEN: <https://quizlet.com/472541354/learn>, VAN OLYAN HOGY EGYEZŐ A KÉRDÉS DE ELTÉRŐ A VÁLASZ.

A kernel egyik alapvető feladata a felhasználói módban működő taszkok felügyelete	I
Az időosztásos rendszerek egyben multiprogramozottak is.	I
A FreeRTOS egy valós idejű, beágyazott környezetben használt operációs rendszer.	I
A valós idejű rendszerek egy eseményre adott időn belül biztosan reagálnak.	H
Kliens (PC, telefon, tablet) gépeken több Unix-alapú rendszer fut, mint windows alapú.	I
A rendszerszolgáltatások védett üzemmódban működnek.	H
A rendszerszolgáltatások védett módban működnek	H
A kernel az első program, ami a háttértárról betöltve a processzor futtatni kezd	H
A Winlogon előbb fut, mint az SMSS (munkamenet-kezelő) a Windows-on.	H
A rendszerhívások jellemzően (de nem mindig) megszakítással járnak együtt.	I
A taszkok minden adminisztratív adatát a kernel címterében tároljuk.	H
A szál egy olyan taszk megvalósítás, amely egy folyamaton belül más szálakkal közös vermet használ.	H
Az Apache webszerver szálalapú változata nagyobb teljesítményre (kérés / mp) képes, mint a folyamatalapú	I
Az exec() Unix rendszerhívás elindít egy új folyamatot.	H
A körülfordulási idő egyenlő a várakozási és a futási idő összegével.	I
A virtuális és fizika memóriacímek futás idejű transzformációja alapvetően szoftveres úton történik.	H
A laphiba oka jellemzően az alkalmazás hibás viselkedése.	H
A laphiba kezelése általában taszkok közötti kontextusváltással is jár.	I
A kerettáblák és laptáblák száma megegyezik	H
A malloc() rendszerhívás kiadása nem jelenti azt, hogy a kernel memóriakezelője azonnal foglal (?) egy további keretet a taszk számára	I
Az operációs rendszer elszeparálja (védi) egymástól a taszkokat, ezért azok védett módban futnak.	H
Az időosztásos operációs rendszerek egyben multi-programozottak is.	I
A FreeBSD egy linux disztribúció.	H
A Windows 10 operációs rendszer mobiltelefonon is működik.	I

A valósidejű rendszerek helyes működés esetén mindig adott határidőn belül válaszolnak az eseményekre	H
A <b>Válaszidő</b> azt fejezi ki, hogy a felhasználónak mennyit kellett várnia egy program első válaszára (kimenetére) annak elindításától számítva.	H
A taszkok adminisztratív adatait a kernel a saját címtartományában tárolja.	H
A Unix exec() rendszerhívás betölt és elindít egy új folyamatot	H
A szál egy olyan taszk, amely egy folyamaton belül más szálakkal közös virtuális memóriaterületet, de saját vermet használ.	I
Futó állapotból futásra kész állapotba csak preemptív ütemezés esetén válhatnak a taszkok.	H
A legrövidebb hátralévő löketidejű (SRTF) ütemező preemptív.	I
A Windows Task Scheduler egy hosszú tavú ütemező.	I
A körforgó (Round-Robin) ütemezés esetén a konvoj-hatás nem jelentkezik, mivel az ütemező a taszkokat csak adott ideig futtatja, utána átütemezéssel megszakítja a futásukat.	I
A többszintű visszacsatolt sorok (MFQ) ütemező preemptív.	I
A kernel átütemezési pontok alkalmazása javítja a válaszidőt, mivel védett módban is teljesen preemptív válik az OS működése	H
Két egymástól független taszk azonos virtuális címen általában eltérő adatot lát, de várhatóan lesznek olyan virtuális címeik, amelyeken azonos programkód található.	I
A laphiba azt jelenti, hogy az adott lap ütem létezik sehol, ezért nem lehet rá hivatkozni.	H
A laphiba gyakorisága a CPU-kihasználtsággal lineárisan nő.	H
Az óra és az újabb esély lapcsere ugyanazon múltbeli adatokra támaszkova működik.	I
Ha növeljük egy rendszerben a fizikai memória méretét, akkor mindig csökkenni fog a laphibák száma, hiszen több lapot tarthatunk bent a memóriában.	H
A multiprogramozott operációs rendszer nevét onnan kapta, hogy egyszerre több programozási nyelven is programozható.	H
A beágyazott operációs rendszerek jellemzően előre meghatározott feladatok ellátását támogatják.	I
A Unix egy időosztásos, multiprogramozott operációs rendszer.	I
Minden valósidejű rendszer mindig adott időkereten belül válaszol a bementre.	H

Az OS kernelek minden része (eljárása) védett módban működik.	H
Egy felhasználói feladatot egy taszk old meg.	H
A szál egy szekvenciális működésű taszk, amely az operációs rendszerben (pl. Linux vagy Windows) található többi szállal közös memóriaterületet használ.	H
A zombi állapot célja azt megvárni, hogy a szülő folyamat nyugtázza a gyerek leállítását.	I
Egy nem terhelt (pl. a felhasználó bejelentkezésére várakozó) operációs rendszeren percenként legfeljebb néhány kontextusváltás történik, hiszen csak 1-2 taszk működik.	H
A taszkok minden adminisztratív adatát a kernel címterében helyezik el.	H
A mai operációs rendszerek jellemzően preemptív ütemezőt használnak.	I
A Legrövidebb löketidejű (SJF) algoritmus esetén a futásra készvé vált taszk beillesztésének művelete $O(1)$ konstans komplexitású.	H
A Körforgó (RR) ütemezőben a taszkok mindig addig vannak futó állapotban, amíg le nem jár az időszeletük.	H
A válaszidő mindig kisebb, mint a körölfordulási idő.	I
Heterogén többprocesszoros ütemezésben a taszkok jellemzően nem migrálhatók szabadon a végrehajtó egységek között.	H
A PRAM modell nem engedi meg a közös memória konkurens írását két (vagy több) taszk által, ezért ilyen esetekben is garantálja a programok helyes működését.	H
A hálózati (socket) kommunikáció egy indirekt, üzenetküldésen alapuló megoldás.	H
A programokban mindenféle jelzésre szabadon beállíthatunk jelzéskezelő eljárásokat.	H

A közös erőforrások használatakor kialakuló versenyhelyzeteket szinkronizációs eszközökkel kezelhetjük.	I
Spinlock (spinning lock) Test-and-set (TSL) művelet segítségével megvalósítható.	I
Az operációs rendszer alapvető célja a hardver eszközök konfigurációja és menedzselése.	H
A beágyazott operációs rendszer egy adott feladatra specializált, valós idejű működésű szoftver	I
A mikrokernelok elosztott felépítésűek, ahol a kernel feladatait alapvetően egymástól független, felhasználói módban futó taszkok oldják meg.	I
A UNIX futási szintje meghatározza a rendszerben futó szolgáltatások körét.	I
Az OS X XNU kernel egy hibrid kernel.	I
A memória-intenzív feladatok I/O-intenzívvé válhatnak, ha sok memóriára van szükségük.	I
A körülfordulási idő az állapotátmeneti gráfban megtett futásra kész – fut – futásra kész kör teljes ideje.	H
Egy multiprogramozott operációs rendszer futtatásához több processzormagra van szükség.	H
A szál egy olyan folyamat, amely más szálakkal közös címtérben fut.	H
Taszkok adminisztratív adatai a kernel és a taszkok címtérében is elhelyezhetők.	I
Az ütemezés során leghatékonyabban láncolt listák segítségével tarthatjuk nyilván taszkok (pl. prioritás szerint) rendezett halmazát.	H
A legrégebben várakozó (FCFS) ütemezési algoritmus FIFO adatstruktúrát használ	I

A körforgó ütemezés nem kooperatív, és elkerüli a kiéheztetést	I
A legrövidebb löketidejű (SJF) algoritmus konstans komplexitású.	H
A többszintű visszacsatolt sorok ütemező (MFQ) az I/O-intenzív taszkokat magas prioritású szinteken szolgálja ki	I
A PRAM (pipelined RAM) modell írás-olvasás ütközésnél mindig először az írás műveletet hajtja végre, hogy az olvasás már az új értékkel térhessen vissza.	H
Aszinkron üzenetváltásos kommunikáció során a küldés művelet befejezése megelőzi fogadás művelet elindítását.	H
A szinkronizáció a taszkok működésének összehangolása a művelet-végrehajtás időbeli korlátozásával.	I
A mutex többpéldányos erőforrások védelmére alkalmas szinkronizációs eszköz.	H
A holtponthoz elleni védekezés legjobb módja a strucc algoritmus	H
A mai operációs rendszerek monolitikus felépítésűek és modulárisan bővíthetők	H
A beágyazott operációs rendszerek minden esetben determinisztikus működésűek.	H
Az időosztásos operációs rendszer alkalmazása csökkenti a rendszer válaszidejét a klasszikus multiprogramozott rendszerekhez képest.	I
Minden rendszerhívás védett módban hajtódik végre.	H
A valósidejű rendszer mindig egy adott időkorláton belül válaszol az eseményekre.	H
A várakozási idő a taszk futásra kész állapotban eltöltött ideje.	H
A szál egy szekvenciális működésű taszk, amely egy folyamaton belül más szálakkal közös virtuális memóriát használ.	I
A legrégebben várakozó (FCFS) ütemező konstans algoritmus komplexitású.	I
Körforgó ütemezés során egy taszk csak akkor lép ki a futó állapotból, ha lejárt az időszelete.	H

A konvoj hatás például a legrövidebb hátralévő löketidejű (SRTF) algoritmussal kezelhető.	I
PRAM modell szerinti kommunikáció csak egy folyamaton belüli szálak között valósítható meg a mai operációs rendszerekben.	H
Az üzenetsor egy indirekt kommunikációs megoldás.	I
A hálózati kommunikáció (socket) egy aszimmetrikus kommunikációs forma.	I
A távoli eljárás hívás megvalósítása csak egy folyamaton belüli szálak között lehetséges, mivel azok férnek hozzá egymás memóriaterületéhez és eljárásaihoz	H
A UNIX jelzések a kommunikáció leggyorsabb formái közé tartoznak nagyon alacsony késleltetésük miatt.	H
Olyan rendszerben, ahol nem alakulhat ki versenyhelyzet, nincs szükség szinkronizációra	H
A szinkronizáció általában rontja programjaink teljesítményét	I
A spinlock kis rezsiköltséggel rendelkező zárolási eszköz, ezért minden esetben javasolt a használata, amikor az operációs rendszer azt támogatja	H
Ha egy rendszerben kialakulhat holtpon, akkor az az erőforrás-foglalások tetszőleges sorrendje esetén ki fog alakulni.	H
Optimista zárolás alkalmazásával minden esetben javítható a programunk teljesítménye.	H
Egy operációs rendszer nem lehet egyszerre monolitikus és moduláris felépítésű.	H
A kemény valósídejű rendszer helyes működés esetén mindig egy adott időkorláton belül válaszol az eseményekre.	I
Egy rendszer hívás meghívása jellemzően szoftveres megszakítást von maga után.	I
A modern mikrokernelek (pl. L4) fő gyengesége a lassú üzenetalapú kommunikáció.	H
A futási szint (runlevel) meghatározza a UNIX rendszerekben futó taszkok prioritását.	H
Az ütemező átbocsájto képessége a taszkok által időegység alatt átvitt adatok mennyisége.	H
A taszkok adminisztratív adatait védelmi okokból mindig a kernel címterében tároljuk.	H
A memória-intenzív taszkok nagy memórafoglalás esetén CPU-intenzívvé válnak.	H
A mai operációs rendszer kernelek eseményvezérelt működésűek.	I
NUMA architektúrájú rendszerben a taszkok nem érik el az összes fizikai memóriát.	H
Nem preemptív ütemező esetén egy taszk kizárólag önszántából veszítheti el a CPU-t.	H

Ha egy rendszerben csak I/O-intenzív taszkok vannak, akkor a legrégebben várakozó (FCFS) ütemező alkalmazása során nem léphet fel konvoj-hatás.	I
A körforgó (RR) ütemező használata optimális átlagos várakozási időt eredményez.	H
A legrövidebb löketidejű (SJF) ütemező adatműveleti konstans komplexitásúak.	H
A mai operációs rendszerek többprocesszoros ütemezési megoldási jellemzően szimmetrikus rendszereket, azaz minden végrehajtó egység ellátja a saját ütemezését.	I
A PRAM modell nem biztosít kölcsönös kizárást a közösen használt memóriaterületre.	H
Az üzenetváltásos kommunikáció során mindig szükséges az átvitt adatok átmeneti tárolása.	H
Az rpcgen egy kódgenerátor, amely RPC interfészleírásból bináris programkódot készít.	H
A szinkronizáció megvalósítása szükségképpen maga után vonja a taszkok várakoztatását.	H
Olyan rendszerben is szükség lehet szinkronizációra, ahol nem alakulhat ki versenyhelyzet	I
A Unix cron egy középtávú ütemező.	H
A Windows Task Scheduler egy hosszú távú ütemező.	I
Az ütemező a várakozó állapotú taszkok közül választja ki a következő futó taszkot	H
Az ütemező átbocsájto képessége az egységnyi időszlet alatt átütemezett taszkok száma.	H
A körülfordulási idő egyenlő a várakozási és a futási idők összegével. A körforgó (Round-Robin) ütemezés esetén a konvoj-hatás nem jelentkezik, mivel az ütemező a taszkokat csak adott ideig futtatja, utána átütemezéssel megszakítja a futásukat.	I
Az ideális ütemező komplexitása lineáris, azaz $O(N)$ .	H
A konvoj hatás kooperatív ütemező algoritmussal nem kezelhető. ----- VÁLASZ: Egyes esetekben kezelhető, másokban nem.	
Egy általános célú operációs rendszerben jellemzőek 1-2 kontextusváltás történik másodpercenként	H
Egy rendszer absztrakt virtuális gépei összességükben több erőforrást tartalmaznak, mint amennyi fizikailag rendelkezésre áll.	I
Egy taszk futásra kész állapotból várakozó állapotba is átkerülhet.	H
Egy taszk futó állapotból futásra kész állapotba csak preemptív ütemezés esetén kerülhet át.	H
Egy feladatot mindig egy taszk old meg.	H
A szálaknak saját verme van.	I

A többszintű visszacsatolt sorok (MFQ) ütemező bemeneti szintválasztó algoritmus a taszkok prioritása alapján határozza meg a kezdeti szintjüket.	H
A statikus többszintű ütemezőkben nem jelentkezhethet a konvoj-hatás, hiszen a globális ütemező preemptív.	H
Egy statikus többszintű ütemező a kimenti szintválasztó algoritmus lehet körforgó (round-robin) ütemező.	I
A többszintű visszacsatolt sorok (MFQ) ütemező az I/O intenzív taszkokat magasabb prioritási szinten tartja, mint a CPU-intenzíveket.	I
A statikus többszintű ütemezőkben nem jelentkezhethet a kiéheztetés, hiszen a globális ütemező preemptív.	H
A többszintű visszacsatolt sorok (MFQ) ütemező preemptív.	I
Az IEEE POSIX egy szabvány, amely előírja a kernel belső felépítését.	H
Egy mikrokernél alapvetően elosztott rendszer felépítésű.	I
A rendszerkönyvtárak az operációs rendszer védett módban működő részei.	H
Az ablakkezelő (window manager) a karakteres parancsértelmező (shell) grafikus változata.	H
A szál a taszk egy olyan megvalósítása, amely önálló memóriaterülettel rendelkezik.	H
A kiéheztetés statikus prioritást alkalmazó prioritásos ütemezőkben nem kerülhető el.	I
A folyamatokon belül csak egy verem lehet.	H
A kooperatív ütemezők nem veszik el a futó taszktól a processzort.	I
A konvoj-hatás például a legrövidebb hátralevő löketidejű (SRTF) algoritmussal megszüntethető.	I
Egy többszintű ütemező akkor is lehet preemptív, ha minden szinten kooperatív ütemezési algoritmust használ.	I
Kooperatív ütemező esetén a taszkok nem hajtanak végre fut -> futásra kész állapotátmenetet	H
A mai operációs rendszerekben jellemzően többféle ütemező működik egyszerre	I
VÁLASZTÓ	
Egy operációs rendszerben a felhasználói alkalmazások védett módban működnek, felettük a felhasználói módban működő kernel gyakorol felügyeletet.	H
A vergődés a laphibák túlzott gyakorisága, amely a rendszer kis mértékű teljesítményromlásával jár.	H



UNIX rendszerekben a felhasználói programok által kiadott rendszerhívások jellemzően a libc rendszerkönyvtár közreműködésével hajtódnak végre.	I
Az NTFS fájlrendszer 8+3 karakteres fájlneveket használ.	H
Egy rendszerben több laptábla van, mint ahány kerettábla található.	I
Ha magas a laphiba gyakorisága, akkor a rendszerben a memória-intenzív taszkok sok időt töltenek el várakozó állapotban.	I
A mai operációs rendszerek alapvetően előretekintő lapozást alkalmaznak, mivel az képes előre jóslni, hogy mely lapokra lesz szükség a jövőben, így csökkenti a laphibák számát.	I
A taszkok memóriatartományát egymástól szeparálni kell, hogy ne zavarják egymás működését.	I
Előfordulhat, hogy több különböző taszkhoz tartozó virtuális memóriacím ugyanarra a fizikai címre képződik le.	I
Az újabb esély (SC) lapcsere algoritmus a lapok használatát is figyelembe veszi.	I
Egy operációs rendszerben jellemzően több laptábla van, mint kerettábla	I
A CPU a cserehelyen (swap) levő adatokhoz közvetlenül nem fér hozzá.	I
A copy-on-write technika lehetővé teszi, hogy két taszk ugyanazt a fizikai memóriát használja.	I
Az alkalmazások által használt összes memória mérete sosem haladhatja meg a fizikai memória méretének kétszeresét.	H
A laphiba gyakoriság a fizikai memória méretének növelésével mindig csökken.	H
A virtuális és fizikai memóriacímek futásidejű transzformációja alapvetően szoftveres úton történik.	H
A taszkok már elindulásukkor lefoglalják a számukra szükséges teljes memóriatartományt.	H
Az újabb esély (SC) lapcsere algoritmus a lapok módosítását is figyelembe veszi.	H
A kernel jellemzően akkor szabadít fel egy használatban lévő fizikai memóriakeretet, amikor arra egy új igény teljesítése miatt van szükség.	H
A laphibák száma nagyon sok párhuzamosan működő taszk esetén lineárisan függ a multiprogramozás fokától.	H
A malloc () rendszerhívás azonnal lefoglalja a taszk által igényelt teljes memóriát.	H
Az osztott memória PRAM modell szerint működik.	I
Aszinkron adatátvitel alatt a küldőnek nem kell megvárnia az adatok fogadását.	I
Az RPC kommunikációs rendszer rpcgen programja interfész leírásából forráskódot generál.	I

A POSIX osztott memória az egyik leggyorsabb kommunikációs forma, amely egy gépen belüli két tetszőleges szál között használható.	I
A PRAM modell szerint írás-írás ütközés esetén valamelyik írás eredménye található majd a memóriában, harmadik érték nem alakulhat ki.	I
Ha egy taszk nem tud belépni a kritikus szakaszba, akkor mindig várakozó állapotba kerül.	I
A foglalva várakozás a holtponthoz kialakulásának szükséges és elégséges feltétele.	H
Létezik optimista zárolás hardveres megvalósítását támogató, széles körben elterjedt CPU	I
Az üzenetsor egy indirekt kommunikációs megoldás.	I
A hálózati kommunikáció (socket) egy aszimmetrikus kommunikációs forma.	I
A távoli eljáráshívás megvalósítása csak egy folyamaton belüli szálak között lehetséges, mivel azok férnek hozzá egymás memóriaterületéhez és eljárásaihoz.	H
Olyan rendszerben, ahol nem alakulhat ki versenyhelyzet, nincs szükség szinkronizációra.	H
A szinkronizáció általában rontja programjaink teljesítményét.	I
A spinlock kis rezsiköltséggel rendelkező zárolási eszköz, ezért minden esetben javasolt a használata, amikor az operációs rendszer azt támogatja.	H
Ha egy rendszerben kialakulhat holtponthoz, akkor az az erőforrás-foglalások tetszőleges sorrendje esetén ki fog alakulni.	I
Optimista zárolás alkalmazásával minden esetben javítható a programunk teljesítménye.	H
A szuperblokk a fájlrendszer metaadatait tartalmazza, és jellemzően több példányban tárolják.	I
Az Unix fájlrendszerekben található fájlok többsége direkt blokkcímekkel elérhető.	I
A leggyorsabb adattároló rendszer a merevlemez meghajtó (HDD).	H
A memóriaalapú permanens tárolók (SSD-k) élettartama jellemzően 5 év körüli.	H
A RAID10 megbízhatósága jobb, de teljesítménye gyengébb a RAID1-nél.	H
Ajánlott zárolás (advisory locking) esetén az OS nem biztosít eszközöket a zárolás megvalósítására.	H
A UNIX fájlrendszerekben található fájlok jelentős része direkt blokkcímekkel elérhető a többszörös indirekciót alkalmazó index címtáblában.	I
A blokkgyorsítótár a fizikai memóriát használja a diszkműveletek gyorsítására	I
A Ceph alapját képező RADOS tárolási rendszer képes az adatok tárolási redundanciát futásidőben, leállás nélkül megváltoztatni.	H

A Windows FAT láncolt listás adatblokk allokációt használ.	I
A Windows 10 alatt a felhasználó saját fájljai a \Windows könyvtárban található.	H
A RAID6 megbízhatósága jobb a RAID1-nél.	I
Windows 10 alatt a telepített felhasználói programok a \Windows könyvtárba kerülnek.	H
Nem szükséges biztonsági mentés akkor, ha az adatokat megfelelően redundáns tárolórendszerben helyezük el, hiszen az véd a rendszerhiba miatti adatvesztés ellen.	H
Ajánlott (advisory) zárolás esetén egy taszk figyelmen kívül hagyhatja egy fájl zárolását.	I
A Windows PowerShell egy grafikus parancsértelmező.	H
Az UNIX ps parancs fájlok listázására szolgál.	H
Az UNIX X11 egy kijelzőszerver protokoll, amely Windows alatt is elérhető a megfelelő alkalmazás telepítése után.	I
Az ablakkezelő feladata az alkalmazásablakok elhelyezésének és megjelenésének vezérlése.	I
A bash egy karakteres parancsértelmező.	I
A legrövidebb előre (SJF) ütemező a gyakorlatban nehezen alkalmazható.	I
Az ütemezők nem szálakat, hanem folyamatokat ütemeznek.	H
A körforgó ütemezés képes elkerülni a kiéheztetést.	I
Az SJF tekinthető prioritásos ütemezőnek is.	I
A mai operációs rendszerek alapértelmezett módon a kemény affinitást alkalmazzák a többprocesszoros ütemezésben	H
A mai operációs rendszer kernelek alapvetően eseményvezérelt működésűek	I
Az átlagos várakozási idő kisebb az átlagos körülfordulási időnél.	I
A memória-intenzív taszkok túl nagy memórafoglalás esetén CPU-intenzívvé válhatnak.	I
Az SJF ütemező preemptív.	H
A kiéheztetés jelensége preemptív ütemezőkben nem alakulhat ki, hiszen megszakíthatják a CPU-t sokáig használó taszk futását.	I
A konvoj-hatás a legrövidebb hátralevő löketidejű (SRTF) algoritmussal elkerülhető.	I
A Windows ütemezője képes elkerülni taszkok kiéheztetését.	I
A többszintű visszacsatolt sorok (MFQ) ütemező prioritásos és preemptív.	I

Elterjedt operációs rendszerekben nincs lehetőség a taszkokat felfüggesztett állapotokba helyezni.	H
A szál olyan taszk, amely más szálakkal közösen osztozhat egy folyamat virtuális memóriatartományán, vermén (stack) és halomján (heap).	H
Az I/O-intenzív taszkok kontextusváltásainak száma jellemzően nagyobb a CPU-intenzív taszkokénál.	I
Dinamikus többszintű ütemezők esetén az I/O-intenzív taszkok átlagos körülfordulási ideje kisebb, mint az átlagos várakozási idejük.	H
FCFS (FIFO) ütemezés esetén jelentkezhethet a konvoj-hatás, ami pl. preemptív prioritásos ütemezésre váltással kezelhető.	I
A várakozási idő a taszk futásra kész állapotban eltöltött ideje.	H
A szál egy szekvenciális működésű taszk, amely egy folyamaton belül más szálakkal közös virtuális memóriát használ.	I
A legrégebben várakozó (FCFS) ütemező konstans algoritmikus komplexitású.	I
Körforgó ütemezés során egy taszk csak akkor lép ki a futó állapotából, ha lejárt az időszelete.	H
A mikrokernelben lényegesen kevesebb védett módban futó programrész található, mint a monolitikus kernelekben	I
„Binary translation” CPU virtualizáció esetén a virtuális gép monitor (VMM) futásidőben cseréli le a taszkok programkódjának egyes utasításait.	I
Az IEEE POSIX operációs rendszerek felhasználói és programozói interfészeit szabványosítja.	I
A moduláris felépítésű kernelek akár működésük közben (futásidőben) is képesek lehetnek egyes kernelmodulok betöltésére vagy eltávolítására.	I
A mikrokernel koncepció egyik célja a védett módban futó kernel részek méretének csökkentése, ezért leglényegesebb előnyük a működési hatékonyságuk növekedése	I
A Unix rendszerek indulásakor alapvetően az init folyamat helyezi üzembe az OS magasabb szintű szolgáltatásait.	I
A mai operációs rendszerek monolitikus felépítésűek és modulárisan bővíthetők.	I
Minden rendszerhívás védett módban hajtódik végre.	I
A valós idejű rendszer mindig egy adott időkorláton belül válaszol az eseményekre.	I
A mai operációs rendszerek jellemzően többszintű ütemezést használnak.	I
A mikrokernel elosztott felépítésűek, ahol az egyes kernel taszkok közös memórián keresztül kommunikálnak.	I

Egy operációs rendszer forráskódja akár több tízmillió programsorból is állhat.	I
Beágyazott operációs rendszerekben esetenként az OS és az alkalmazás nem különül el, egyetlen végrehajtható fájlba fordul.	I
A systemd egy sysinit alternatíva.	I
A rendszerkönyvtárak az operációs rendszer védett módban futó eljárásait tartalmazó programkönyvtárak.	H
"Bare-metal" virtualizáció esetén a vendég OS ütemezője dönt a CPU kiosztásáról a virtuális gépek között.	H
A rendszerkönyvtárak eljárásai védett módban működnek.	H
A beágyazott operációs rendszerek minden esetben determinisztikus működésűek.	H
A rendszerhívások meghívása minden esetben üzemmódváltással jár együtt.	H
A IaaS egy olyan megoldás, amely felhasználói módban futó szolgáltatásokat virtualizál.	H
Egy moduláris kernel nem lehet monolitikus.	H
Az osztott memória egy PRAM (pipelined RAM) modell szerint működő kommunikációs eszköz.	I
A jelzések szinkron kommunikációt valósítanak meg a küldő és a fogadó között.	H
PRAM (pipelined RAM) modell szerinti kommunikáció csak egy folyamaton belüli szálak között valósítható meg a mai operációs rendszerekben.	H
A távoli eljáráshívás (RPC) számítógépek között is működik, nemcsak egy gépen belül.	I
Sok olvasó és kevés író esetén a szemafor nem a legalkalmasabb szinkronizációs eszköz.	I
Egypéldányos erőforrások esetén a holtponthoz az erőforrás-allokációs gráfban detektálható.	I
Ha egy rendszerben kialakulhat holtponthoz, akkor az az erőforrás-foglalások tetszőleges sorrendje esetén ki fog alakulni.	H
Lehetséges várakozásmentes I/O műveletek alkalmazása a programjainkban.	I
A RAID10 megbízhatósága jobb, de teljesítménye gyengébb a RAID1-nél.	H
Ajánlott zárolás (advisory locking) esetén az OS nem biztosít eszközöket a zárolás megvalósítására.	H
A Unix fájlrendszerekben található fájlok jelentős része direkt blokkcímekkel elérhető.	I
Ha letörölök egy szimbolikus link által hivatkozott fájlrendszeri bejegyzést, akkor az adat elvész. Ha magát a szimbolikus linket törölöm le, akkor nem.	I
A RAID 5 egy kijelölt paritástárolót (partíciót, diszket) használ a redundáns tárolás céljára.	H

Megbízható (kellően redundáns) tárolórendszer esetén nincs szükség mentésre, hiszen nem következhet be adatvesztés.	H
A PRAM modellben az írás-olvasás ütközéskor az olvasás mindig a régi értékkel tér vissza	H
Egy folyamaton belüli szálak a PRAM (pipelined RAM) modell szerint tudnak egymással kommunikálni	I
A postaláda (mailbox) a PRAM modell szerint működik, hiszen mindenki ugyanahhoz az adattárolóhoz tér hozzá	H
Az osztott memória egy PRAM (pipelined RAM) modell szerint működő kommunikációs eszköz	I
Optimista zárolás alkalmazásával minden esetben javítható a programunk teljesítménye	H
A holtpont detektálására és kezelésére nem minden esetben érdemes felkészítenünk a programunkat	?
A mai operációs rendszerek monolitikus felépítésűek és modulárisan bővíthetők.	H
A beágyazott operációs rendszerek minden esetben determinisztikus működésűek.	H
Az időosztásos operációs rendszer alkalmazása csökkenti a rendszer válaszidejét a klasszikus multiprogramozott rendszerekhez képest.	I
Minden rendszerhívás védett módban hajtódik végre.	H
A valósídejű rendszer mindig egy adott időkorláton belül válaszol az eseményekre.	H
A várakozási idő a taszk futásra kész állapotban eltöltött ideje.	H
A szál egy szekvenciális működésű taszk, amely egy folyamaton belül más szálakkal közös virtuális memóriát használ.	I
A legrégebben várakozó (FCFS) ütemező konstans algoritmikus komplexitású.	I
Körforgó ütemezés során egy taszk csak akkor lép ki a futó állapotából, ha lejárt az időszelete.	H
A konvoj hatás például a legrövidebb hátralevő löketidejű (SRTF) algoritmussal kezelhető.	I
PRAM modell szerinti kommunikáció csak egy folyamaton belüli szálak között valósítható meg a mai operációs rendszerekben.	H
Az üzenetsor egy indirekt kommunikációs megoldás.	I
A hálózati kommunikáció (socket) egy aszimmetrikus kommunikációs forma.	I
A távoli eljárás-hívás megvalósítása csak egy folyamaton belüli szálak között lehetséges, mivel azok férnek hozzá egymás memóriaterületéhez és eljárásaihoz.	H

A UNIX jelzések a kommunikáció leggyorsabb formái közé tartoznak nagyon alacsony késleltetésük miatt.	H
Olyan rendszerben, ahol nem alakulhat ki versenyhelyzet, nincs szükség szinkronizációra.	H
A szinkronizáció általában rontja programjaink teljesítményét.	I
A spinlock kis rezsiköltséggel rendelkező zárolási eszköz, ezért minden esetben javasolt a használata, amikor az operációs rendszer azt támogatja.	H
Ha egy rendszerben kialakulhat holtpon, akkor az az erőforrás-foglalások tetszőleges sorrendje esetén ki fog alakulni.	H
Optimista zárolás alkalmazásával minden esetben javítható a programunk teljesítménye.	H
Egy rendszerhívás meghívása jellemzően szoftveres megszakítást von maga után.	I
A futási szint (runlevel) meghatározza a Unix rendszerekben futó taszkok prioritását.	H
A mai operációs rendszer kernelek alapvetően eseményvezérelt működésűek.	I
Az átlagos várakozási idő kisebb az átlagos körülfordulási időnél.	I
A memória-intenzív taszkok túl nagy memórafoglalás esetén CPU-intenzívvé válhatnak.	H
NUMA architektúrájú rendszerben a taszkok nem érik el az összes fizikai memóriát.	H
Az SJF ütemező preemptív.	H
A kiéheztetés jelensége preemptív ütemezőkben nem alakulhat ki, hiszen megszakíthatják a CPU-t sokáig használó taszk futását.	I
A konvoj-hatás a legrövidebb hátralevő löketidejű (SRTF) algoritmussal elkerülhető.	I
A Windows ütemezője képes elkerülni taszkok kiéheztetését.	I
A többszintű visszacsatolt sorok (MFQ) ütemező prioritásos és preemptív.	I
A Windows FAT láncolt listás adatblokk allokációt használ.	I
A Windows 10 alatt a felhasználó saját fájljai a Windows könyvtárban találhatóak.	H
A RAID6 megbízhatósága jobb a RAID1-nél.	I
Az IEEE POSIX operációs rendszerek felhasználói és programozói interfészeit szabványosítja.	I
A rendszerkönyvtárak eljárásai védett módban működnek.	H
A moduláris felépítésű kernelek akár működésük közben (futásidőben) is képesek lehetnek egyes kernelmodulok betöltésére vagy eltávolítására.	I

A mikrokernel koncepció egyik célja a védett módban futó kernel részek méretének csökkentése, ezért leglényegesebb előnyük a működési hatékonyságuk növekedése.	I
A Unix rendszerek indulásakor alapvetően az init folyamat helyezi üzembe az OS magasabb szintű szolgáltatásait.	I
Elterjedt operációs rendszerekben nincs lehetőség a taszkokat felfüggesztett állapotokba helyezni.	H
A szál olyan taszk, amely más szálakkal közösen osztozhat egy folyamat virtuális memóriatartományán, vermén (stack) és halomján (heap).	H
Az I/O-intenzív taszkok kontextusváltásainak száma jellemzően nagyobb a CPU-intenzív taszkokénál.	I
Dinamikus többszintű ütemezők esetén az I/O-intenzív taszkok átlagos körülfordulási ideje kisebb, mint az átlagos várakozási idejük.	H
FCFS (FIFO) ütemezés esetén jelentkezhethet a konvoj-hatás, ami pl. preemptív prioritásos ütemezésre váltással kezelhető.	I
Egy operációs rendszerben jellemzően több laptábla van, mint kerettábla.	I
A laphibák száma nagyon sok párhuzamosan működő taszk esetén lineárisan függ a multiprogramozás fokától.	H
A CPU a cserehelyen (swap) levő adatokhoz közvetlenül nem fér hozzá.	I
A malloc () rendszerhívás azonnal lefoglalja a taszk által igényelt teljes memóriát.	H
A copy-on-write technika lehetővé teszi, hogy két taszk ugyanazt a fizikai memóriát használja.	I
Windows 10 alatt a telepített felhasználói programok a \Windows könyvtárba kerülnek.	H
Nem szükséges biztonsági mentés akkor, ha az adatokat megfelelően redundáns tárolórendszerben helyezjük el, hiszen az véd a rendszerhiba miatti adatvesztés ellen.	H
Ajánlott (advisory) zárolás esetén egy taszk figy... hagyhatja egy fájl zárolását.	I
Egy operációs rendszer forráskódja akár több tízmillió programsorból is állhat.	I
Egy operációs rendszerben a felhasználói alkalmazások védett módban működnek, felettük a felhasználói módban működő kernel gyakorol felügyeletet.	H
Beágyazott operációs rendszerekben esetenként az OS és az alkalmazás nem különül el, egyetlen végrehajtható fájlba fordul.	I
A systemd egy sysinit alternatíva.	I
Egy moduláris kernel nem lehet monolitikus.	H
A Windows PowerShell egy grafikus parancsértelmező.	H



Az UNIX ps parancs fájlok listázására szolgál.	H
Az UNIX X11 egy kijelzőszerver protokoll, amely Windows alatt is elérhető a megfelelő alkalmazás telepítése után.	I
Az ablakkezelő feladata az alkalmazásablakok elhelyezésének és megjelenésének vezérlése.	I
A bash egy karakteres parancsértelmező.	I
A legrövidebb előre (SJF) ütemező a gyakorlatban nehezen alkalmazható.	I
Az ütemezők nem szálakat, hanem folyamatokat ütemeznek.	H
A körforgó ütemezés képes elkerülni a kiéheztetést.	I
Az SJF tekinthető prioritásos ütemezőnek is.	I
A mai operációs rendszerek alapértelmezett módon a kemény affinitást alkalmazzák a többprocesszoros ütemezésben.	H
Az alkalmazások által használt összes memória mérete sosem haladhatja meg a fizikai memória méretének kétszeresét.	H
Egy rendszerben több laptábla van, mint ahány kerettábla található.	I
Ha magas a laphiba gyakorisága, akkor a rendszerben a memória-intenzív taszkok sok időt töltenek el várakozó állapotban.	I
A laphiba gyakoriság a fizikai memória méretének növelésével mindig csökken.	H
A mai operációs rendszerek alapvetően előretekintő lapozást alkalmaznak, mivel az képes előre jóslni, hogy mely lapokra lesz szükség a jövőben, így csökkenti a laphibák számát.	I
A mai operációs rendszerek jellemzően többszintű ütemezést használnak.	I
A rendszerhívások meghívása minden esetben üzemmódváltással jár együtt.	H
A rendszerkönyvtárak az operációs rendszer védett módban futó eljárásait tartalmazó programkönyvtárak.	H
A mikrokernelek elosztott felépítésűek, ahol az egyes kernel taszkok közös memórián keresztül kommunikálnak.	I
A IaaS egy olyan megoldás, amely felhasználói módban futó szolgáltatásokat virtualizál.	H
A taszkok memóriatartományát egymástól szeparálni kell, hogy ne zavarják egymás működését.	I
Előfordulhat, hogy több különböző taszkhhoz tartozó virtuális memóriacím ugyanarra a fizikai címre képződik le.	I
A virtuális és fizikai memóriacímek futásidejű transzformációja alapvetően szoftveres úton történik.	H

Az újabb esély (SC) lapcsere algoritmus a lapok használatát is figyelembe veszi.	I
A taszkok már elindulásukkor lefoglalják a számukra szükséges teljes memóriatartományt.	H
Az osztott memória PRAM modell szerint működik.	I
Aszinkron adatátvitel alatt a küldőnek nem kell megvárnia az adatok fogadását.	I
Az RPC kommunikációs rendszer rpcgen programja interfész leírásából forráskódot generál.	I
A POSIX osztott memória az egyik leggyorsabb kommunikációs forma, amely egy gépen belüli két tetszőleges szál között használható.	I
A szinkronizáció általában a teljesítmény javulását okozza.	H
A szuperblokk a fájlrendszer metaadatait tartalmazza, és jellemzően több példányban tárolják.	I
Az Unix fájlrendszerekben található fájlok többsége direkt blokkcímekkel elérhető.	I
A leggyorsabb adattároló rendszer a merevlemez meghajtó (HDD).	H
A memórialapú permanens tárolók (SSD-k) élettartama jellemzően 5 év körüli.	H
A RAID10 megbízhatósága jobb, de teljesítménye gyengébb a RAID1-nél.	H
A rendszerkönyvtárak az operációs rendszer védett módban futó eljárásait tartalmazó programkönyvtár.	H
A mikrokernelben lényegesen kevesebb védett módban futó programrész található, mint a monolitikus kernelekben	I
UNIX rendszerekben a felhasználói programok által kiadott rendszerhívások jellemzően a libc rendszerkönyvtár közreműködésével hajtódnak végre.	I
„Bare-metal” virtualizáció esetén a vendég OS ütemezője dönt a CPU kiosztásáról a virtuális gépek között.	H
„Binary translation” CPU virtualizáció esetén a virtuális gép monitor (VMM) futásidőben cseréli le a taszkok programkódjának egyes utasításait.	I
Előfordulhat, hogy több különböző taszkhoz tartozó virtuális memóriacím ugyanarra a fizikai címre képződik le.	I
A virtuális és fizikai memóriacímek futásidejű transzformációja alapvetően szoftveres úton történik.	H
A vergődés a laphibák túlzott gyakorisága, amely a rendszer kis mértékű teljesítményromlásával jár.	H
Az újabb esély (SC) lapcsere algoritmus a lapok módosítását is figyelembe veszi.	H

A kernel jellemzően akkor szabadít fel egy használatban lévő fizikai memóriakeretet, amikor arra egy új igény teljesítése miatt van szükség.	H
A PRAM modell szerint írás-írás ütközés esetén valamelyik írás eredménye található majd a memóriában, harmadik érték nem alakulhat ki.	I
Aszinkron adatátvitel alatt a küldőnek nem kell megvárnia az adatok fogadását.	I
Ha egy taszk nem tud belépni a kritikus szakaszba, akkor mindig várakozó állapotba kerül.	I
A foglalva várakozás a holtponthoz kialakulásának szükséges és elégséges feltétele.	H
Létezik optimista zárolás hardveres megvalósítását támogató, széles körben elterjedt CPU	I
Az NTFS fájlrendszer 8+3 karakteres fájlneveket használ.	H
Ajánlott zárolás (advisory locking) esetén az OS nem biztosít eszközöket a zárolás megvalósítására.	H
A UNIX fájlrendszerekben található fájlok jelentős része direkt blokkcímekkel elérhető a többszörös indirekciót alkalmazó index címtáblában.	I
A blokkgyorsítótár a fizikai memóriát használja a diszkműveletek gyorsítására	I
A Ceph alapját képező RADOS tárolási rendszer képes az adatok tárolási redundanciát (danciat) futásidőben, leállás nélkül megváltoztatni.	H?
A fájlrendszerek felépítése mindig egy fa struktúrával reprezentálható (a gyökér könyvtárból kiindulva a fájlok és a könyvtárak egy fát alkotnak).	H
A leggyorsabb 1TiB méretű adattároló rendszer a merevlemez meghajtó (HDD).	H
A RAID0 általában gyorsabb a RAID1-nél. de a RAID1 megbízhatóbb.	I
Az iSCSI (internet SCSI) fájlokon elvégzett műveletek hálózati átviteli protokollja.	H
A Unix programokban mindenféle jelzésre szabadon beállíthatunk jelzéskezelő eljárásokat	H
Üzenetváltásos kommunikációban a Küldés és a Fogadás szinkron műveletek.	H
A PRAM (pipelined RAM) modell nem engedi meg a közös memória konkurens írását két (vagy több) taszk által, ezért ilyen esetekben is garantálja a programok helyes működését.	H
A kritikus szakasz védelme szemaforral megvalósítható	I