

# Elméleti Zárthelyi dolgozat

**Határidő** Nincs megadva határidő**Pont** 30**Kérdések** 30**Elérhető** máj 4, 11:00-ig**Időkorlát** 25 perc

## Instrukciók

A dolgozat 50%-os eredménytől sikeres.

15-18 pont : 2

19-22 pont : 3

23-26 pont : 4

27-30 pont : 5

Ezt a kvízt ekkor zárolták: máj 4, 11:00.

## Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	<a href="#">1. próbálkozás</a>	25 perc	14 az összesen elérhető 30 pontból

❗ A helyes válaszok el vannak rejtve.

Ezen kvíz eredménye: **14** az összesen elérhető 30 pontból

Beadva ekkor: máj 4, 10:26

Ez a próbálkozás ennyi időt vett igénybe: 25 perc

**Helytelen**

### 1. kérdés

**0 / 1 pont**

A számítógépek fejlődése során megkülönböztetünk számítógépes, operációs rendszer generációkat. Mi az ami leginkább befolyásolta ezen generációkat, azok kialakulását?

☒ A CPU kernel-user szintjének megjelenése.

☐

Sokféle szempont alapján lehet rendszereket megkülönböztetni, de leginkább a Single-Task, Multi-Task rendszerek fejlődése.



A külvilág igénye. Például az első generációknál leginkább számításigényes megoldásokra volt szükség, így azt támogatták.



Sokféle szempont alapján lehet rendszereket megkülönböztetni, de leginkább az elektronikai támogatás fejlődése.

**Helytelen****2. kérdés****0 / 1 pont**

Az operációs rendszerek és a hardverek kapcsolatában egyre nagyobb szerepe van a Firmware-nek. Tényleg így van, mi ez?



Igen, a klasszikus külvilággal folytatott adatcsere (írás-olvasás) kérésre ma már jellemzően nem közvetlenül az eszköz (például merevlemez) válaszol, hanem egy beépített program.



Az operációs rendszerek réteges felépítésűek. Ez a hatás tükröződik vissza, a Firmware megjelenésében is, ez a kernel legújabb rétege!



Igen, az operációs rendszer részévé vált ez az újabb szoftver réteg is.



Ez csak egy elnevezés, a merevlemez ugyanúgy csatlakozik ma a rendszerhez, mint régen.

**Helytelen****3. kérdés****0 / 1 pont**

A háttértáron elhelyezkedő alkalmazások, programoknak és a memóriában futó folyamatok között van-e különbség?



Igen. Ahogy egy úszó a medence partján partravetett hal, úgy a program a zsebben semmi!

- ☒ Van különbség, de csak monoprogramozási környezetben!
- ☐ Nincs különbség, ez csak szómágia!
- ☐ Igen, de csak elnevezésbeli a különbség, ahogy például a Pistát néha Pityunak vagy akár Istvánnak is lehet nevezni!

Helytelen

**4. kérdés****0 / 1 pont**

A perifériákkal végzett kommunikációra melyik alábbi állítás nem érvényes?

- ☐ Folyamatosan érdeklődünk, hogy például elkezdődött-e a mérkőzés?
- ☐ Akár speciális eszközt is biztosíthatunk nagy mennyiségű adat(memória) eléréshez.
- ☐ Az eszközök adatai fájlban gyűlnek, így az operációs rendszer bináris fájl műveleteket biztosít ehhez!
- ☒ Berúgjuk az ajtót, megszakítjuk a futó dolgokat és jelezzük, "meccsidő" van!

Helytelen

**5. kérdés****0 / 1 pont**

Mit biztosít, mit jelent, hogy a CPU üzemmódok bővültek, és megjelent például a "kernel" és "user" mód?

- ☐ Ez a lehetőség azt jelenti, hogy az operációs rendszer tud a hatalmával élni és "teljhatalmú úr" a folyamatok fölött.
- ☐ Olyan lehetőséget ad, amivel mi is készíthetünk Operációs rendszert!



Olyan lehetőséget ad, amivel jobban igazítható egy-egy alkalmazástípushoz a rendszerünk.



Ez a lehetőség azt jelenti, hogy az operációs rendszer nem tud a hatalmával élni és nem lehet "önkényúr" a folyamatok fölött.

**Helytelen****6. kérdés****0 / 1 pont**

Mi a szerepe a lemezkérés ütemezéseknek?



Optimális sávszélesség elérése a háttértárak felé.



Az interleave helyes beállítása.



A lemez forgási sebességét csökkenteni vagy növelni a hatékonyság, hozzáférési idő csökkentésében.



Az író/olvasó fej optimális működését biztosítani.

**Helytelen****7. kérdés****0 / 1 pont**

Melyik lemez ütemezési algoritmust célszerű használni SSD háttértárak esetén?



Az SSTF algoritmust.



A C-SCAN algoritmust.



A SCAN algoritmust.



Az FCFS algoritmust.

**8. kérdés****1 / 1 pont**

Mint jelent a háttértárak megbízhatósága?

- ☐ Egy módszer, ami a sávszélességet növeli.
- ☒ Egy módszert, amivel a tárolási redundanciát növelni lehet.
- ☐ Egy módszer, ami a hatékonyságot növeli.
- ☐ Módszer ami biztosítja, hogy a közös adatokat egyszerre csak egy folyamat tudja használni.

**9. kérdés****1 / 1 pont**

Milyen szerepe van ma a mágnesszalagos háttértárnak?

- ☐ Mivel nagyon drága, ezért már nem használják.
- ☐ Mivel lineáris adatelérést biztosít, végig kell tekerni a szalagot a kívánt állományig, lassú, ezért nem használt.
- ☐ Nagygépes környezetek elengedhetetlen perifériája az LTO6-os egység.
- ☒ Lecsökkent a szerepe, nagy mennyiségű adat, megbízható tárolását biztosítja.

**10. kérdés****1 / 1 pont**

Mi az i-node (index-node) tábla?



A Unix stílusú operációs rendszerek fájlrendszerének alapstruktúrája, táblája. Egy node tartalmazza a fájl attribútumokat és a fájl blokkok helyét a lemezen.



Csak a legújabb Linux stílusú operációs rendszerek fájlrendszerének alapstruktúrája, táblája. Egy node tartalmazza a fájl attribútumokat és a fájl blokkok helyét a lemezen.



Egy általános index tábla, ami a fájlrendszer indexelését tartalmazza.



A Windows operációs rendszerek fájlrendszerének alapstruktúrája, táblája. Egy node tartalmazza a fájl attribútumokat és a fájl blokkok helyét a lemezen.

### 11. kérdés

1 / 1 pont

Egy háttértáron hány különböző fájlrendszert használhatunk egyidejűleg?



Ahány egységre, partícióra bontjuk a háttértárat.



Ahány fájlrendszert az operációs rendszer ismer.



Kettőt.



Csak egyet.

Helytelen

### 12. kérdés

0 / 1 pont

Milyen kötelező kiegészítő információkat, attribútumokat kell tárolni egy fájlról vagy könyvtárról?



Nincs ilyen, emiatt lehet hidden fájl vagy hidden fájlrendszer is.



Van ilyen, az IEEE2020 szabvány definiálja.



Nincs ilyen általános előírás, de minden fájlrendszer típus pontosan előírja, hogy hol, hogyan és milyen információkat tárol az adott bejegyzésről.



Van ilyen ISO szabvány.

Helytelen

### 13. kérdés

0 / 1 pont

A preemptív operációs rendszer modell "mit előz meg, kitől kit véd meg"?



A határidők megjelenése arra ösztönzi az alkalmazásokat, hogy gyorsan fejezzék be a válaszadásokat, így biztosítva a rendszert.



A modellben bizonyos idő után megkéri az alkalmazást, hogy mondjon le a CPU-ról, így megelőzi a rendszer kisajátítását, összeomlását.



A modellben bizonyos idő után elveszi a rendszer a vezérlést az alkalmazástól, így megelőzi a rendszer kisajátítását, összeomlását.



Nem előz meg semmit, hiszen a mai rendszerekben is találkozunk rendszerhibákkal.

### 14. kérdés

1 / 1 pont

A Peterson féle kölcsönös kizárás használatánál mi a probléma az alábbi kódrészlet piros sorai felcserélése során?

```
void belepes(int proc)
```

```
{int masik;
```

```
    masik=1-proc; //mivel N=2...
```

```
    akarja[proc]=1; //processz futni akar
```

```
    kovetkezo=proc;
```

```
while( kovetkezo==proc &&  
      akarja[masik]);  
}
```



Ha bármikor vált az ütemező "éhen nem halhatnak a filozófusok", nincs probléma.



Ha rosszkor vált az ütemező "egymásnak eshetnek a filozófusok", többen is kritikus szakaszba kerülhetnek.



Ha a sorok előtt vált az ütemező az problémát okozhat.



Ha a sorok után vált az ütemező, az okozhat problémát.

### 15. kérdés

1 / 1 pont

Mi a problémája a tevékeny várakozással megvalósított kölcsönös kizárásnak? (PI, Peterson algoritmus)



Ez a megoldás ideális, nem véletlen, hogy a mai rendszerek "SpinLock" néven is használnak hasonlót.



Ha erőforrás szűkében vagyunk, akkor ez rossz, mert CPU időt pazarol.



Ez a megoldás nem jó, mert többen is kritikus szekcióba kerülhetnek.



Ha erőforrás szűkében vagyunk, akkor ez rossz, mert memóriát pazarol.

### 16. kérdés

1 / 1 pont

A szekvenciális folyamatmodell alapján melyik állapotban lehet csak egy folyamat?





Futásra kész állapotban, mert így lesz egyértelmű, hogy ki lesz a következő "futó".



Blokkolt állapotban, mert egy blokkolt folyamat a következőt is blokkolná ha ebbe az állapotba szeretne bekerülni.



Futó állapotban.



Mindegyikbe lehet egy folyamat, ha nincs sok futó alkalmazás a rendszerben.

### 17. kérdés

1 / 1 pont

Mi a szemafor? (informatikában használt)



Olyan speciális változó, aminek segítségével alvás-ébredés alapú kölcsönös kizárást tudunk biztosítani.



Olyan felhasználói szintű objektum, aminek segítségével alvás-ébredés alapú kölcsönös kizárást tudunk biztosítani.



A felhasználói programok által foglalt összes memóriát ezzel a speciális változóval lehet kezelni!



Olyan speciális változó, amit csak a kernel használ automatikusan az alvás-ébredés alapú kölcsönös kizárás megvalósítására.

Helytelen

### 18. kérdés

0 / 1 pont

Mi a prioritásos ütemezők alapelve és az ebből következő probléma?



Felváltva választunk a magas majd az alacsony prioritási osztályból, így nincs probléma.



Mindig a magasabb prioritású folyamat következik, ami biztosítja a helyes működést.



Mindig az alacsonyabb prioritású folyamat következik, és így megakadályozzuk a kiéheztetést.



Az alapelv, ha van magasabb prioritású folyamat ami CPU-ra vár, azt fogja választani az ütemező, így az alacsony prioritású folyamatra a kiéheztetés veszélye fenyeget.

## 19. kérdés

1 / 1 pont

Mi az üzenetküldéssel biztosított kölcsönös kizárás fő előnye?



Nem szükséges közös (rendszer) memória a helyes kölcsönös kizárás biztosításához.



Az, hogy a hálózati kommunikációhoz hasonló elvet követ.



Közös (rendszer) memória esetén nincs lehetőség a helyes kölcsönös kizárás biztosítására.



Még a monitornál is könnyebb használat.

## 20. kérdés

1 / 1 pont

Mivel javítják az alapértelmezett prioritásos ütemezési elvet, hogy elkerüljék a kiéheztetést?

- ☒ Valamilyen prioritási osztályok közti átmenetek megvalósításával.
- ☐ Nem javítják semmivel.
- ☐ Azzal, hogy minden felhasználói folyamat 0-ás prioritással fut.
- ☐ Azzal, hogy egy prioritási osztályban Round-Robin ütemezést valósítunk meg.

**21. kérdés****1 / 1 pont**

A Fogyasztó-Vásárló probléma megoldásában, miért probléma ha felcseréljük a két pirossal jelölt sort?

```
void pék()
{
    int kenyér;
    while (1)
    {
        kenyér=pék_süt();
        down(&üres);
        down(&szabad);
        kenyér_polcra(kenyér);
        up(&szabad);
        up(&tele);
    }
}
```

Mert ha az ütemező a

kenyér\_polcra(kenyér);

- ☐ utasítás előtt vált, holtpont alakulhat ki.
- ☒ Mert a kölcsönös kizárás úgy hibás lehet.
- ☐ Nem probléma ha felcseréljük a két sort.

Mert ha az ütemező a

```
kenyér=pék_süt();
```

- ☐ utasítás után vált, holtpont alakulhat ki.

Helytelen

## 22. kérdés

0 / 1 pont

Jelölje meg mi nem a Sorsjáték ütemezés jellemzője?



Prioritásos ütemezés, ami azt jelenti, hogy ennek megfelelően osztja szét a folyamatok között a sorsjegyeket.



Sorsjáték szerűen a kiosztott jegyek alapján választunk egy folyamatot.



Hasonlít a Garantált ütemezéshez, csak a CPU idő számolás más elvek alapján történik.



Az időintervallum lejártakor egy új sorsjegyet kap a folyamat és legközelebb rövidebb ideig futhat.

Helytelen

## 23. kérdés

0 / 1 pont

Virtuális memóriakezelés során mi a címfordítás?



A programok a virtuális térben helyezkednek el, így a virtuális címből fizikai címet kell előállítani.



A programok a virtuális térben helyezkednek el és az első szabad fizikai címhez rendelést hívjuk címfordításnak.



A programok a virtuális térben helyezkednek el, így annak megadása, hogy melyik lapon van az aktuális kódrész.



A program a fizikai memóriában fut, és ebből a címből meg kell határozni a virtuális címet.

**Helytelen****24. kérdés****0 / 1 pont**

Mikor nem teljes folyamat mozgatása történik a memória és háttértár között?



Multiprogramozás rögzített memória szeletek használata esetén.



Multiprogramozás memória csere használata esetén.



Soha nincs és nem is volt ilyen.



Virtuális memória kezelés esetén, csak extrém esetben lehet ilyen.

**25. kérdés****1 / 1 pont**

Mi a virtuális memória kezelés?



Nincs ilyen, hiszen egy CPU csak a fizikai memóriában lévő kódot tudja végrehajtani.



Virtuális gépen végzett memória kezelés.



Mikor a létező fizikai memóriától nagyobb memóriát is engedélyez használni az operációs rendszer.



A SWAP partíció használatával megvalósuló memória kezelés.

**26. kérdés****1 / 1 pont**

Melyik feladat nem tartozik a memória kezelő (Memory Management Unit, MMU) hatáskörébe virtuális memóriakezelés során?

- ☐ A Data Execution Prevention (DEP) beállítása.
- ☐ A Copy On Write (COW) beállítása.
- ☐ A virtuális laptábla kezelése.
- ☒ A program betöltésekor szükséges LDT, GDT táblák módosítása.

**Helytelen****27. kérdés****0 / 1 pont**

Mi a megszakítás(interrupt) fogalma?



A megszakítás kérés jellemzően I/O eszközökhöz kötődik, egy csapda, amivel azt jelzi, hogy kernel módba kell váltani!



A megszakítás kérés nem kötődik I/O eszközökhöz, egyszerűen egy gépi kódú utasítás.



A megszakítás kérés jellemzően I/O eszközökhöz kötődik, egy időzítő jelzés küldése a CPU felé, amiben azt jelzi, hogy lejárt az adatérvényesség ideje.



A megszakítás kérés jellemzően I/O eszközökhöz kötődik, egy valós jelzés küldése a CPU felé, ami azt jelzi, hogy szeretné az eszköz ha vele foglalkozna a CPU!

**28. kérdés****1 / 1 pont**

### Mi az Input-Output eszközök feladata?

- ☐ Nincs is feltétlenül szükség I/O eszközökre egy számítógép működéshez!
- ☐ Megbízhatóan ütemezze az adatok fogadását és küldését.
- ☒ A számítógép és külvilág kapcsolatának a biztosítása.
- ☐ Mivel a számítógép digitális eszköz, ezért az analóg-digitális jelátalakítás a feladata.

Helytelen

### 29. kérdés

0 / 1 pont

Mi a probléma figyelmen kívül hagyásának (strucc algoritmus) lényege?

- ☒ Ütemezzük a kéréseket a NOOP (No Operation) szerint.
- ☐ Nem foglalkozunk az eszközök felé irányuló kérések ütemezésével, csak biztosítjuk a kérések küldését.
- ☐ Nem foglalkozunk az eszközök felé irányuló kérések ütemezésével, a kérések küldését az alkalmazás közvetlenül végzi az eszközhöz.
- ☐ Ütemezzük a kéréseket a CDF (Closest Device First) algoritmus szerint.

Helytelen

### 30. kérdés

0 / 1 pont

Melyik program réteg nem tartozik az eszköz és az operációs rendszer kapcsolata közé?

- ☐ Eszköz függő operációs rendszer interfész.

- ☐ Felhasználói I/O könyvtár.
- ☐ Megszakításkezelés rétege
- ☒ Eszköz független operációs rendszer interfész.

Kvízeredmény: **14** az összesen elérhető 30 pontból