

Név:	Neptun kód:
------	-------------

I. Teszt kérdések

Σ / 30 pont

Útmutató:

Jelölje egyértelműen a helyes választ! Karikázza be az I-t, ha az állítás igaz. Karikázza be a H-t, ha az állítás hamis. Karikázza be a ?-et, ha nem tudja a választ. Ha javítani akar a válaszon, akkor húzza át mind a három mezőt, és írja a sor végére a választ (Igaz/Hamis/Nem tudom).

Pontozás: Helyes válasz 1 pont, rossz válasz -1 pont. Kérdéscsoportonként a minimum pont 0 pont.

1. kérdéscsoport. A processzor ütemezéssel kapcsolatos kérdések.

1.	Az ütemező algoritmus feladata a FUTÁSRA KÉSZ állapotban lévő feladatok közül kiválasztani a FUTÓ állapotba kerülő feladatot. Ezt a feladatot az algoritmusnak valós időben kell megoldania.	I	H	?	
2.	Az SRTF algoritmus az SJF algoritmus olyan preemptív változata, amely egy periodikus óraütésre újraütemezi a FUTÁSRA KÉSZ folyamatokat, és a legrövidebb löketidővel rendelkezőt választja ki futásra.	I	H	?	Az SRTF algoritmus az SJF algoritmus preemptív változata, amely egy feladat beérkezése esetén újraütemezi a FUTÁSRA KÉSZ folyamatokat, és a legrövidebb maradék löketidővel rendelkezőt választja ki futásra.
3.	A többszintű sorok alkalmazása esetén a FUTÁSRA KÉSZ állapotba kerülő feladatokat azok prioritása alapján sorokba helyezük. A CPU ütemezés során már csak a legmagasabb prioritású, feladatot tartalmazó sorból választunk egy feladatot, és azt helyezük FUTÓ állapotba.	I	H	?	
4.	Többszintű sorok alkalmazása esetén a kisebb prioritású feladatok kiéheztetése léphet fel, ha a magasabb prioritású feladatok folyamatosan elhasználják a teljes processzoridőt, de ez egy kerülendő állapot.	I	H	?	

2. kérdéscsoport. Feladatok leírása.

5.	A JAVA virtuális gép egy folyamat a hoszt operációs rendszeren belül. A JAVA szálak feleltethetők meg a hoszt operációs rendszer szálainak, ez többnyire one-to-one (JAVA szál egyben OS szál is) napjainkban.	I	H	?	
6.	Egyes operációs rendszerekben csak szálak hozhatók létre. Ezek az operációs rendszerek nem alkalmazhatók biztonságkritikus területeken, mert nem való-	I	H	?	A biztonság kritikusság és a szálak között semmi kapcsolat nincs.

	sul meg a feladatok közötti szeparáció.				
7.	A coroutine-ok alkalmazása esetén a kölcsönös kizárás megoldása a programozó feladata egy szálon belül futó coroutine-ok esetén, ez mindig megoldható a yield hívások megfelelő elhelyezésével.	I	H	?	
8.	A coroutine lényegében azonos a fiber-rel, de míg a coroutine programnyelvi szintű elem, addig a fiber rendszer szintű szolgáltatás.	I	H	?	

3. kérdéscsoport. Prioritás inverzió és elkerülése.

9.	A prioritás inverzió egy speciális holtpon, amely során egy vagy több közös erőforráson keresztül alacsonyabb prioritású feladatok feltartanak egy nagyobb prioritású feladatot.	I	H	?	A prioritás inverzió nem holtpon.
10.	A prioritás öröklés során egy erőforrást használó alacsonyabb prioritású feladat véglegesen megkapja az adott erőforrásra várakozó legmagasabb prioritású feladat prioritását.	I	H	?	Csak az éppen végrehajtott kritikus szakasz végéig kapja meg.
11.	A prioritás plafon protokoll futási időben történő alkalmazása garantálja, hogy nem fordulhat elő prioritás inverzió a rendszerben.	I	H	?	A prioritás plafon protokoll tervezési időben használható.
12.	A prioritás öröklés és a prioritás plafon protokollok az asztali és szerver operációs rendszerben választható konfigurációs opciók.	I	H	?	Beágyazott operációs rendszerekben szokott választható opció lenni.

4. kérdéscsoport. Holtpon és kezelése.

13.	Egy rendszer feladatainak egy H részhalmaza holtponon van, ha a H halmazba tartozó valamennyi feladat olyan eseményre vár, amelyet csak egy másik, H halmazbeli feladat tudna előállítani.	I	H	?	
14.	A holtpon előállításának a szükséges feltételei a következők: Kölcsönös kizárás, foglalva várakozás, erőszakos erőforrás elvétel, körkörös várakozás.	I	H	?	Az erőszakos erőforrás elvétel esetén nem lehet holtpon, a szükséges feltétel az, hogy éppen hogy ne legyen erőszakos erőforrás elvétel.
15.	A holtpon észlelése és feloldása akkor kísérhető meg, ha az erőforrások és a feladatok visszaállítási pontokkal rendelkeznek, vagyis holtpon esetén a holtpon előtti helyzetükbe visszaállíthatók.	I	H	?	
16.	A bankár algoritmus a holtpon megelőzés egyik módszere, amely lehetővé teszi, hogy csak azok a feladatok futhassanak, amelyek erőforrás igényei biztosan kielégíthetőek.	I	H	?	A bankár algoritmus egy holtpon elkerülési algoritmus, a holtpon megelőzési algoritmusok tervezési idejű algoritmusok.

5. kérdéscsoport. A Windows ütemezésével kapcsolatos kérdések.

17.	Windowsban a kernel módú szálak prioritása változhat a futás során.	I	H	?	Kernel módúnak is lehet dinamikus (1-15) prioritása.
18.	Windowsban a 16-31 közötti prioritású szálak kemény valós idejű viselkedésűek.	I	H	?	Nem, ezek nem valós idejűek, csak nem változhat a prioritásuk futás közben.
19.	Windowsban a szálakhoz a processzor prioritáson kívül I/O prioritás is rendelhető.	I	H	?	Windowsban a szálakhoz a processzor prioritáson kívül I/O prioritás is rendelhető.
20.	Windowsban a kernel szálakhoz tartozó ütemezési időszeket hosszabb, mint a felhasználói szálaké.	I	H	?	Vagy ugyanolyan hosszú minden szálé, vagy az előtérben futó program szálai a hosszabbak

6. kérdéscsoport. Virtualizációval kapcsolatos kérdések.

21.	Egy bare-metal virtualizációs megoldás esetén a hypervisor feladata, hogy a hardver erőforrások kiosztását elvégezze.	I	H	?	Ezért bare-metal.
22.	Egy virtualizációs megoldástól elvárt a Popek és Goldberg féle követelmények alapján, hogy a virtuális gépeken futtatott programok ugyanazt az eredményt adják, mintha fizikai gépen futtatnánk őket.	I	H	?	Azonosság követelmény.
23.	CPU virtualizáció esetén tiszta emuláció használatával jó teljesítményt lehet elérni.	I	H	?	Pont, hogy ez a hátránya.
24.	A <i>trap and emulate</i> CPU virtualizációs módszer működéséhez processzor támogatás is kell.	I	H	?	Privilegizált műveleteknek trap-et kell kiváltaniuk.

7. kérdéscsoport. Permanens tár kezelése.

25.	A fájl (file) a permanens táron az adattárolás logikai egysége. Az operációs rendszer feladata a logikai egységek (fájlok) leképzése valódi fizikai egységekre (blokkokra).	I	H	?	
26.	Az alacsony szintű fájlrendszer egyik feladata a fizikai blokkok cache-elése, ezt a modern asztali és szerver operációs rendszerekben egy dedikált puffer cache-sel oldják meg.	I	H	?	Modern operációs rendszerekben a lapozással összevont módon, a virtuális memóriával oldják meg.
27.	A Linux esetén szinte az összes szabad fizikai memória keretet felhasználja a rendszer diszk cache-ként.	I	H	?	

8. kérdéscsoport. SAN és NAS technológiák.

28.	A SAN technológiák felhasználásával blokk szinten oszthatunk meg permanens tároló eszközöket a hálózaton. Pl. ilyen technológia a Fibre Channel, az iSCSI, az AoE, és az NFS.	I	H	?	Az NFS NAS technológia.
29.	A NAS technológiák fájl szinten teszik elérhetővé a szerveren tárolt fájlokat. Pl. ilyen technológia a SMB/CIFS és a HTTP.	I	H	?	A HTTP fájl átviteli technológia, nem fájlmegosztás.
30.	SAN-on, pl. iSCSI-n, elérhetővé tett permanens tárról lehetséges az operációs rendszer indítása, de ehhez firmware (pl. BIOS) és operációs rendszer támogatás is szükséges.	I	H	?	

II/1. Nagykérdés

Σ / 10 pont

Egy rendszerben az alábbi feladatok találhatóak:

Feladat	Beérkezési idő (ms)	CPU löket (ms)
P1	0	6
P2	3	21
P3	3	3
P4	5	6

a) Adja meg a várakozási idő definícióját és mértékegységét is! (1 pont)

Várakozási idő (waiting time): Az összes idő, amit a feladat várakozással töltött. Mértékegység: s (másodperc)

$$t_{\text{waiting}} = t_{\text{ready}} + t_{\text{other, non-running}}$$

b) Az SJF (3 pont) és Round-robin (5 pont) ütemező algoritmusok közül melyik nyújtja a kisebb átlagos várakozási időt a megadott terhelésnél? Válaszában adja meg pontosan, hogy melyik feladat mikor fut (Gantt diagram). A Round-robin esetén az időszelvény 4 ms. A megoldásban mutassa be, hogyan jutott az eredményre (köztes lépések, futásra kész sor elemei, stb. jelenjenek meg)!

SJF:

Időintervallum (s)	Futó feladat	Várakozó feladatok
1-3	P1	-
3-5	P1	P2, P3
5-6	P1	P2, P3, P4
6-9	P3	P2, P4
9-15	P4	P2
15-36	P2	-

Várakozási idő: $t_{w,p1}=0$ ms $t_{w,p2}=15-3=12$ ms $t_{w,p3}=6-3=3$ ms $t_{w,p4}=9-5=4$ ms $t_w=(0+12+3+4)/4=(19/4)$ ms=4.75ms

RR: Zárójelben a még megmaradt CPU löketet írrom be

Időintervallum (s)	Futó feladat	Várakozó feladatok (FIFO)
0-3	P1	-
3-4	P1	P2(21),P3(3)
4-5	P2	P3(3),P1(2)
5-8	P2	P3(3),P1(2),P4(6)
8-11	P3	P1(2),P4(6),P2(17)
11-13	P1	P4(6),P2(17)
13-17	P4	P2(17)
17-21	P2	P4(2)
21-23	P4	P2(13)
23-27	P2	-
27-31	P2	-
31-35	P2	-
35-36	P2	-

Várakozási idő: $t_{w,p1}=0+7=7$ ms $t_{w,p2}=1+9+2=12$ ms $t_{w,p3}=5=5$ ms $t_{w,p4}=8+4=12$ ms $t_w=(7+12+5+12)/4=(36/4)$ ms=9ms

c) Melyik ütemezőt használjuk a gyakorlatban, röviden indokolja a választát (1 pont)?

A Round-robin ütemezőt, elsősorban más ütemezési algoritmusok részeként. Bár az SJF tulajdonságai jobbak, de a gyakorlatban a CPU löketet nem tudjuk előre, vagyis az algoritmus elsősorban elméleti jelentőségű. Ugyanakkor az RR egyszerűen megvalósítható egy HW időzítő megszakítással.

II/2. Nagykérdés

Σ / 10 pont

a) Mutassa be a megadott adatokkal, példákon keresztül a változó méretű memória partíciók lefoglalásánál használt first fit, next fit, best fit, és worst fit algoritmusokat. A helyes foglalási lépések megadása algoritmusonként 2 pont értékű. (8 pont összesen)

Egy rendszerben az adott pillanatban 23 Mbyte, 64 Mbyte, 11 Mbyte, 79 Mbyte, 12 Mbyte, 50 Mbyte és 40 Mbyte méretű szabad területek vannak. Hogyan fog a fenti négy algoritmus sorrendben a 65 Mbyte, 22 Mbyte, 48 Mbyte, 12 Mbyte, 62 Mbyte méretű memória igénynek helyet foglalni?

Abba a szabad területbe, ahol allokáltam, beírtam a maradékot, és szürke háttérrel jelöltem, sajnos a hallgatók sokféle jelölést alkalmazhat...

First fit:

Foglalás	23M	64M	11M	79M	12M	50M	40M
65M				14M			
22M	1M						
48M		16M					
12M		4M					
62M	Nem allokálható, mert nincs ekkora összefüggő szabad terület						

Next fit:

Foglalás	23M	64M	11M	79M	12M	50M	40M
65M				14M			
22M						28M	
48M		16M					
12M		4M		2M			
62M	Nem allokálható, mert nincs ekkora összefüggő szabad terület						

Best fit:

Foglalás	23M	64M	11M	79M	12M	50M	40M
65M				14M			
22M	1M						

48M						2M	
12M					0M		
62M		2M					

Minden igény kielégíthető.

Worst fit:

Foglalás	23M	64M	11M	79M	12M	50M	40M
65M				14M			
22M		42M					
48M						2M	
12M		30					
62M	Nem allokálható, mert nincs ekkora összefüggő szabad terület						

b) Indokolja a tapasztaltakat. (2 pont)

Pl.

A "best fit minimális töredezettséggel jár megállapítás", de erőforrás igényes. 1 pont.

A worst fit a legrosszabb, és hasonlóan erőforrás igényes, mint a best fit. 1 pont.