

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. Operációs rendszerek típusai, felépítése.

1.	Személyi számítógép (PC) felhasználásával nem lehet beágyazott rendszert építeni, ahhoz speciális hardver és operációs rendszer kell (pl. nem Linux vagy Windows).	I	H	?	Az alkalmazástól függ, nem az alkalmazott SW vagy HW komponensektől.
2.	A védelmi szintek a CPU-ban az alacsony szintű erőforrásokhoz történő hozzáférést szabályozzák.	I	H	?	
3.	Az operációs rendszerekben a felhasználó direkt kapcsolatban van az alkalmazói programokkal az I/O eszközökön (billentyűzet, egér, VGA) keresztül.	I	H	?	Az OS-en keresztül kapcsolódnak.
4.	A mikrokernél alapú operációs rendszerek erőforrás igényesebbek, ugyanakkor védettek az eszközmeghajtók és a nem alapvető kernel szolgáltatások hibáitól, mivel azok eltérő védelmi szinten futnak.	I	H	?	

2. kérdéscsoport. Processzor ütemezéssel kapcsolatos kérdések.

5.	Preemptív ütemezést alkalmazó operációs rendszerekben bizonyos operációs rendszer feladatok nem szakíthatók meg.	I	H	?	
6.	Többszintű prioritásos sorokat használó ütemező esetén egy szinten mindig round-robin (körforgó) ütemezést alkalmaznak.	I	H	?	Leggyakrabban azt használják, de lehet például FIFO, vagy akármi más is.
7.	A többprocesszoros rendszerekben alkalmazott u.n. self-scheduling megoldás lényegében azt jelenti, hogy minden processzor a saját munkáját ütemezi a rendszerben.	I	H	?	

8.	A kemény processzor affinitás megadása azért szükséges SMP rendszerekben, mert egyébként a CACHE koherencia fenntartása miatt csökkenne a teljesítmény.	I	H	?	Az affinitásra ezért van szükségünk, hogy a CACHE találati arány magas legyen.
----	---	---	---	---	--

3. kérdéscsoport. Üzenet alapú kommunikáció. Holtpont.

9.	A postaláda mindig megszűnik az őt létrehozó felhasználói folyamat terminálódása során.	I	H	?	Lehet OS szintű is a postaláda...
10.	A nem blokkoló hívás esetén az eredmények (tényleges visszatérési érték) és a mellékhatások a hívás visszatérése után jelentkeznek. A hívás visszatérési érték csupán a hívásra vonatkozó státuszinformáció.	I	H	?	
11.	Az RPC hívás során a hívó fél passzívan várakozik a hívás lefutására.	I	H	?	
12.	A holtpont meglétének szükséges feltétele, hogy a rendszerben ne legyen u.n. "erőszakos erőforrás elvétel".	I	H	?	A 4 feltétel közül az egyik feltétele.

4. kérdéscsoport. Kölcsönös kizárással kapcsolatos kérdések.

13.	Folyamatok között megvalósuló kölcsönös kizárás megvalósítására alkalmazható a lock bit.	I	H	?	Nem, mert a lock bit csak közös memóriában képzelhető el, az meg nincs folyamatok esetén.
14.	Egy többpéldányos erőforrásból egy folyamatnak 4 példányra van szüksége a futáshoz. Ebben az esetben a 4 erőforrás egymás után egyenként lefoglalható (pl. egy for ciklussal) hibamentesen.	I	H	?	Nem, mert holtponthoz vezethet ez a megoldás. A 4 példányt egyetlen kérdésben kell lefoglalni.
15.	A mutex lényegében egy bináris szemafornek tekintethető.	I	H	?	
16.	Két tetszőleges szál közötti kommunikációra alkalmazható a heap.	I	H	?	Csak egy folyamat kontextusában futó szálak esetén.

5. kérdéscsoport. Memóriakezeléssel kapcsolatos kérdések.

17.	Az általános célú operációs rendszerekben logikai címet a központi egység (CPU) generálja a folyamat futása közben, majd ezeket képzile az MMU (ha van) fizikai címekre.	I	H	?	
18.	A dinamikusan beszerkesztett programkönyvtárak (pl. Windows	I	H	?	

	DLL) több program számára is elérhetőek (code sharing).				
19.	A Translation look-aside buffer (TLB) feladata a logikai címek fizikai címekre történő leképzése.	I	H	?	A leképzés folyamatának a gyorsítása a feladata. A kérdés nem elég egyértelmű...
20.	A virtuális memória alkalmazása esetén egy program végrehajtásához nem szükséges a teljes programkód betöltése.	I	H	?	Ez az egyik oka, a virtuális memória alkalmazásának.

6. kérdéscsoport. UNIX operációs rendszerekkel kapcsolatos kérdések.

21.	A UNIX két tradicionális alapváltozata a BSD és a System V.	I	H	?	
22.	A folyamatok adminisztratív adatai a kernel címterében vannak.	I	H	?	Hamis, mivel az u-terület a folyamat címterében van.
23.	A fork() rendszerhívás sikeres végrehajtás esetén 0 értékkel tér vissza.	I	H	?	Hamis, mivel szülő esetén a gyerekek PID-jét adja vissza.
24.	UNIX jelzések csak szülő-gyerek viszonylatban működnek.	I	H	?	Hamis, jelzést más, pl. a felhasználó és a kernel is küldhet.

7. kérdéscsoport. Felhasználó-és jogosultságkezelés témakörrel kapcsolatos kérdések.

25.	A biztonság fogalma a sértetlenség és a bizalmasság fogalmak egy időben történő megvalósulását jelenti.	I	H	?	Még a rendelkezésre állásnak is meg kell valósulnia.
26.	A hitelesítés során azt vizsgáljuk, hogy a rendszerhez intézett kérést valójában ki küldte, a következő fázisban az engedélyezés során azt vizsgáljuk, hogy a hitelesített személy mit tehet a rendszerben.	I	H	?	
27.	A UNIX operációs rendszerekben csak rendszergazda jogosultsággal olvasható és írható a /etc/passwd file. Így védik a jelszavakat.	I	H	?	Bárki olvashatja, nem itt tárolják a jelszavat.

8. kérdéscsoport. uC/OS-II operációs rendszerrel kapcsolatos kérdések.

28.	A uC/OS-II preemptív, valós-idejű beágyazott operációs rendszer, amely tisztán C programozási nyelven került megírásra.	I	H	?	Minden igaz, kivéve az, hogy részben Assembly nyelven van írva.
29.	A uC/OS-II ütemezője prioritásos, és egy prioritási szinten egy feladat (task) futtatható csak.	I	H	?	

30.	A uC/OS-II operációs rendszert az alkalmazás indítja el.	I	H	?	
-----	--	---	---	---	--

II/1. Nagykérdés

Σ / 10 pont

Egy rendszerben az alábbi feladatok találhatók:

Feladat	Beérkezési idő (ms)	CPU löket (ms)
P1	0	3
P2	3	24
P3	3	3
P4	5	6
P5	8	3

A FIFO és az SRTF ütemező algoritmusok közül melyik nyújtja a kisebb átlagos körülfordulási időt az adott terhelésnél? Adja meg a körülfordulási idő definícióját és mértékegységét is! A megoldásban mutassa be, hogyan jutott az eredményre!

Megoldás: Körülfordulási idő: Egy feladatra vonatkozóan a rendszerbe helyezéstől a teljesítésig eltelt idő.

$$t_{CPU, végrehajtás} + t_{várakozás}$$

A körülfordulási idő mértékegysége a másodperc (s).

FIFO ütemező:

Időtartam (ms)	FUT	Futást kezdte (ms)	Futást befejezi (ms)	Ready sor tartalma
0-3	P1	0	3	-
3-5	P2	3	27	P3
5-8	P2	3	27	P3,P4
8-27	P2	3	27	P3,P4,P5
27-30	P3	27	30	P4,P5
30-36	P4	30	36	P5
36-39	P5	36	39	-

Feladat	Beérkezési idő (ms)	Kilépési idő (ms)
P1	0	3
P2	3	27
P3	3	30
P4	5	36
P5	8	39

Körbefordulási idő: $t_{ta} = (3+24+27+31+31)/5 = 116/5 = 23,2$ ms

SRTF ütemező:

Időtartam	FUT	Futást kezdte	Futást befejezi	Ready sor tartalma
0-3	P1	0	3	-
0-5	P3	3	-	P2
5-6	P3	3	6	P2,P4
6-8	P4	6	-	P2
8-11	P5	8	11	P2,P4
11-15	P4	6	15	P2
15-39	P2	15	39	-

Feladat	Beérkezési idő (ms)	Kilépési idő (ms)
P1	0	3
P2	3	39
P3	3	6
P4	5	15
P5	8	11

Körbefordulási idő: $t_{ta} = (3+36+3+10+3)/5 = 55/5$ ms = 11 ms

A SRTF ütemező a kedvezőbb a körbefordulási idő szempontjából, ami érhető a P2 által okozott konvoj hatás teljesítményt rontó hatása miatt a FIFO ütemező esetén. Kisebb mértékben a P4 folyamat megszakítása is csökkenti a körbefordulási időt (még az SJF ütemezőhöz képest is, amit

a feladatban nem kellett vizsgálni).

II/2. Nagykérdés

Σ / 10 pont

Egy igény szerinti lapozást használó rendszerben 3 vagy 4 fizikai memórialap áll egy folyamat rendelkezésére. A futás folyamán sorban a következő virtuális lapokra történik hivatkozás:

0, 1, 3, 2, 3, 1, 4, 1, 0, 1, 2, 3, 4

Hány laphiba következik be a rendszerben a következő algoritmusok esetén, ha kezdetben a 3 vagy 4 lap üres?

- legrégebbi lap (FIFO) algoritmus alkalmazásánál 3 vagy 4 fizikai memória lap esetén
- újabb esély algoritmus (Second chance) algoritmus alkalmazásánál 3 és 4 fizikai memória lap esetén

Röviden magyarázza meg az eredményeket!

A megoldásban mutassa be, hogyan jutott az eredményre, csak a laphibák számának megadását nem fogadjuk el!

FIFO 3 memória kerettel:

Lapok	0	1	3	2	3	1	4	1	0	1	2	3	4
FIFO0	0	1	3	2	2	2	4	1	0	0	2	3	4
FIFO1		0	1	3	3	3	2	4	1	1	0	2	3
FIFO2			0	1	1	1	3	2	4	4	1	0	2
Laphiba	I	I	I	I			I	I	I		I	I	I

Laphibák száma: 9

FIFO 4 memória kerettel:

Lapok	0	1	3	2	3	1	4	1	0	1	2	3	4
FIFO0	0	1	3	2	2	2	4	4	0	1	1	3	3
FIFO1		0	1	3	3	3	2	2	4	0	0	1	1
FIFO2			0	1	1	1	3	3	2	4	4	0	0
FIFO3				0	0	0	1	1	3	2	2	4	4
Laphiba	I	I	I	I			I		I	I		I	

Laphibák száma: 8

Újabb esély 3 memória kerettel: Kódolás: lap száma, és a referenced/used bit értéke

Lapok	0	1	3	2	3	1	4	1	0	1	2	3	4
FIFO0	0,y	1,y	3,y	2,y	2,y	2,y	4,y	1,y	0,y	0,y	2,y	3,y	4,y
FIFO1		0,y	1,y	3,n	3,y	3,y	2,n	4,y	1,y	1,y	0,n	2,y	3,y
FIFO2			0,y	1,n	1,n	1,y	3,n	2,n	4,y	4,y	1,n	0,n	2,y
Laphiba	I	I	I	I			I	I	I		I	I	I

Laphibák száma: 9

Újabb esély 4 memória kerettel: Kódolás: lap száma, és a referenced/used bit értéke

Lapok	0	1	3	2	3	1	4	1	0	1	2	3	4
FIFO0	0,y	1,y	3,y	2,y	2,y	2,y	4,y	4,y	0,y	0,y	0,y	3,y	3,y
FIFO1		0,y	1,y	3,y	3,y	3,y	2,n	2,n	1,n	1,y	1,y	0,n	0,n
FIFO2			0,y	1,y	1,y	1,y	3,n	3,n	4,y	4,y	4,y	1,n	1,n
FIFO3				0,y	0,y	0,y	1,n	1,y	2,n	2,n	2,y	4,n	4,y
Laphiba	I	I	I	I			I		I			I	

Laphibák száma: 7

A FIFO esetén a nagyobb memória keretszám ezen terhelésre jobb eredményeket nyújt, de más terhelések esetén a Bélády anomália miatt a memória keretek számának emelése akár a teljesítmény csökkenésével is járhat.

Az újabb esély algoritmus az azonos számú memória kerettel működő FIFO algoritmusnál jobb teljesítményt nyújt, jobban tud élni a terhelés lokalitásából következő előnyökkel.

Az újabb esély algoritmus esetén a nagyobb számú memória keret növeli a teljesítményt.