

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 csoportosítása, tulajdonságai.

1.	A Windows és a Linux nem valós idejű operációs rendszerek, de mindkettőhöz létezik valós idejű működést lehetővé tevő kiegészítés.	I	H	?	Igaz, pl. RTLinux és pl. Ardence RTX.
2.	A Windows és a Linux használható beágyazott rendszerekben.	I	H	?	Igaz, az alkalmazáson és a követelményeken múlik, nem a HW-én vagy az OS-en.
3.	Biztonság kritikus környezetben, pl. járműipar vagy egészségügy, alkalmazott operációs rendszereknek, speciális, alkalmazási kör specifikus minősítéssel kell rendelkezniük.	I	H	?	Igaz.
4.	A valós idejű operációs rendszerek bizonyos operációs rendszer szolgáltatások valós idejű működését garantálják hardware függő időkorlátokkal. Az alkalmazás valós idejű működésének biztosítása az alkalmazás fejlesztőinek a feladata.	I	H	?	Igaz.

2. kérdéscsoport. Virtuális memóriakezelés.

5.	Még virtuális memóriakezelés esetén sem lehet a rendelkezésre álló fizikai memóriánál nagyobb méretű programokat futtatni.	I	H	?	Hamis, elég, ha az aktuális munkakészlet elfér a fizikai memóriába.
6.	Az előretékintő lapozás mindig jobb teljesítményt nyújt, mint az igény szerinti lapozás.	I	H	?	Hamis, csak akkor nyújt jobb teljesítményt, ha jól tudja becsülni a jövőben szükséges lapokat, ha rosszul becsül, akkor csak erőforrást pazarol.
7.	Lapok fizikai memóriába történő fagyasztására például a lapra vonatkozó I/O műveletek miatt, vagy az LFU algoritmus által frissen behozott lapok esetén van szükség.	I	H	?	Igaz.

8.	A vergődés azért is veszélyes, mert azt a hosszú távú ütemező I/O intenzív terhelésnek értelmezheti, és további feladatokat engedhet a rendszerbe a helyzetet tovább rontva.	I	H	?	Igaz.
----	--	---	---	---	-------

3. kérdéscsoport. Windows operációs tulajdonságaival kapcsolatos kérdések.

9.	A Windows operációs rendszerek kombinált szegmens és lapszervezésű memóriakezelést használnak.	I	H	?	Hamis, csak lapszervezés van.
10.	Windows esetén egy épp befejeződött folyamat memórialapjai egyből odaadhatóak egy másik folyamatnak.	I	H	?	Hamis, Előbb ki kell nullázni.
11.	Windowsban laphiba esetén a kért memórialapot mindig a lemezről kell beolvasni.	I	H	?	Hamis. Lehet, hogy még egy standby listán lévő memórialap van.
12.	A Windows a memórafoglalást két lépésben végzi, hogy takarékoskodjon a memóriával.	I	H	?	Igaz, csak a committed mögé kell helyet fenntartani, a reservednek még nem.

4. kérdéscsoport. Windows biztonsággal kapcsolatos kérdések.

13.	A beépített Rendszergazda felhasználó SID-je függ a számítógép SID-jétől.	I	H	?	Igaz, az eleje a gép SID-je.
14.	Windowson egy folyamat jogosultságainak ellenőrzése során mindig a folyamatot elindító felhasználó jogait vizsgálja a rendszer.	I	H	?	Hamis, van "impersonation" is.
15.	Windowsban a registry kulcsok és a fájlok hozzáférés védelme ugyanazon az elven van megvalósítva.	I	H	?	Igaz. Mindegyiknél ugyanolyan ACL-ek vannak.
16.	A Windowsban a Mandatory Integrity Control a védendő objektumok címkézésén alapszik.	I	H	?	Igaz, a MIC úgy működik, hogy a szereplőket és az objektumokat is címkézzük..

5. kérdéscsoport. UNIX folyamatok.

17.	A kernel módba lépett felhasználói folyamatok kernel kontextusba váltanak.	I	H	?	Hamis, mivel folyamat kontextusa nem változik.
18.	Felhasználói módban és kernel kontextusban zajlik a rendszerhívások kiszolgálása.	I	H	?	Hamis, mivel ilyen párosításban nem zajlik semmilyen tevékenység.
19.	A folyamatokat a felhasználó kivonhatja a rövid távú ütemezés alól.	I	H	?	Igaz (felfüggesztett módok).
20.	A folyamatok minden adminisztratív adata a kernel címtérben van.	I	H	?	Hamis, mivel az u-területen is vannak adatok, ami a folyamat címtérben van.

6. kérdéscsoport. UNIX folyamatok kommunikációja.

21.	UNIX jelzések csak szülő-gyerek viszonylatban működnek.	I	H	?	Hamis, jelzést más, pl. a felhasználó és a kernel is küldhet.
22.	A System V IPC üzenetsorokban adattípusokat is használhatunk.	I	H	?	Hamis, csak az üzeneteknek van egyszerű típusa.
23.	A System V osztott memória a leggyorsabb kommunikációs forma.	I	H	?	Igaz, mivel nem igényel kernel közreműködést.
24.	A Sun RPC XDR leírása egy interfész specifikáció.	I	H	?	Igaz.

7. kérdéscsoport. A háttértár kezelése.

25.	A fájlrendszer leképzés egyetlen feladata, hogy megteremtse a kapcsolatot a fizikai blokkok és a fájlok (logikai egység) között.	I	H	?	Hamis, nyilván kell tartania az üres helyeket is.
26.	Az indexelt allokáció esetén a töredezettség mentesítés nem szükséges, hiszen nincs külső tördelődés, csak belső.	I	H	?	Hamis, a töredezettség mentesítés itt a fejmozgás optimalizálására szükséges.
27.	A tranzakció orientált (log-structured, log-based transaction oriented, journaling fájlrendszerek) fájlrendszerek a fájlrendszer konzisztenciáját biztosítják, az adatvesztés elkerülésére nem alkalmasak.	I	H	?	Igaz.

8. kérdéscsoport. Folyamatok, szálak és a hardver összefüggései.

28.	Az operációs rendszerek vagy folyamatok, vagy folyamatok és azokon belül szálak létrehozását támogatják.	I	H	?	Hamis, egyes beágyazott operációs rendszerekben csak szálakat hozhatunk létre.
29.	Folyamatok nem hajthatnak végre I/O műveleteket direkt módon, csak rendszerhívásokon keresztül.	I	H	?	Hamis, rendszerfolyamatok végrehajthatnak I/O műveleteket direkt módon is.
30.	Egyes modern CPU-k MMU-ja lehetővé teszi, hogy memórialapokat megosszunk folyamatok között, többnyire csak olvasásra, de akár írásra is.	I	H	?	Igaz. Lásd pl. read/write bit a laptáblában, DLL és SO code sharing, stb.

Név:	Neptun kód:
------	-------------

II/1. Nagy kérdés

Σ / 10 pont

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

1. sorozat: 0, 1, 2, 3, 0, 1, 2, 3, 4, 1, 2, 3, 4, 1
2. sorozat: 0, 1, 2, 3, 4, 1, 0, 1, 2, 3, 1, 4, 0, 1

Hány laphiba következik be a rendszerben a következő lpcsere algoritmusok esetén a sorozatokra, ha kezdetben a 4 lap üres minden sorozat indításánál?

- Legrégebbi lap (FIFO) algoritmus alkalmazásánál (1-1 pont)
- Újabb esély (SC) algoritmus alkalmazásánál (2-2 pont).

Röviden magyarázza meg az eredményeket! (4 pont)

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

FIFO 4 fizikai memória kerettel és 1. sorozat:

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

Laphibák száma: 5

FIFO 4 fizikai memória kerettel és 2. sorozat:

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

Laphibák száma: 12

SC 4 fizikai memória kerettel és 1. sorozat:

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

Laphibák száma: 5

SC 4 fizikai memória kerettel és 2. sorozat:

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

Laphibák száma: 10

Ha a FIFO teljesen jó, akkor 1 pont, ha az SC teljesen jó a 2 pont. Ha nincs jelölve a used/referenced bit és rossz, akkor 0 pont, ha jelölve van, és láthatóan elszámolta, de tudja az alogitmust (többnyire jó), akkor 1 pont, ha nem tudja az algoritmust akkor 0 pont.

Értékelés:

1. Az első sorozat első részének a munkahalmaza belefér a rendelkezésre álló fizikai memória keretek számába, aztán a munkahalmaz változik, és ezért van egy laphiba (0 lecserélődik 4-re), utána ismét belefér a munkahalmaz a rendelkezésre álló fizikai memória keretekbe. (1 pont)
2. A második sorozat munkahalmaza nem fér bele a rendelkezésre álló fizikai memória keretekbe, de az 1. lapra gyakran hivatkozik a folyamat. (1 pont)
3. 2. sorozatban 2 alkalommal megtakaríthatunk egy lapcserét, mivel az SC algoritmus ezekben az esetekben sikeresen bent tartotta a gyakran használt 1-es lapot. (1 pont)
4. Az SC algoritmus statisztikailag jobban viselkedik (alacsonyabb a laphiba gyakoriság) mint a FIFO, viszont implementációja bonyolultabb, és HW támogatást is igényel (used/referenced) bit. (1 pont)

Ezen kívül lehetnek még értelmes megjegyzések, ezeket többé-kévesbé értelmesen lehet kombinálni is, de most nagyobb dolog nem jut eszembe...

II/2. Nagy kérdés

Σ / 10 pont

Rajzolja le a Windows operációs rendszerek architektúráját a fontosabb felhasználói és kernel módú komponensekkel. (6 pont)

b) Ugyanazon a Windows 7-et futtató számítógépen egy POSIX API-t használó és egy Windows API-t használó alkalmazás is kiad egy-egy fájl megnyitására szolgáló függvényhívást. Ábrázolja, hogy milyen rétegeken keresztül jutnak el a megfelelő rendszerhívás végrehajtásáig, és mikor történik a felhasználói-kernel mód váltás. (A pontos függvény neveket nem kell tudni, csak azt, hogy milyen komponenseken keresztül halad a hívás) (2x2 pont)

POSIX API-t használó

Windows API-t használó

--	--	--