

1. Kako se mijenjaju registri procesora pri izvođenju "obične" instrukcije?

- a) SP (registar stanja) se automatski uvećava nakon dohvata trenutne instrukcije
- b) SP (kazaljka stoga) se automatski uvećava nakon dohvata trenutne instrukcije
- c) PC (programsko brojilo) se automatski uvećava nakon dohvata trenutne instrukcije

2. Izvođenjem instrukcije skoka: B 0x123456 (ili JMP 0x123456) dogodit će se sljedeće:

- a) $SP = 0x123456$
- b) $PC = 0x123456$
- c) $PC = PC + 0x123456$ (mnogi procesori rade "relativan skok")
- d) $PC = SP + 0x123456$

3. Instrukcija za poziv potprograma CALL 0x123456 napraviti će sljedeće:

- a) $SP = 0x123456$
- b) $PC = 0x123456$
- c) spremi PC na stog; $PC = 0x123456$
- d) $PC = 0x123456$; spremi PC na stog

4. Instrukcija za povratak iz potprograma RET napraviti će sljedeće:

- a) SP spremi na stog
- b) PC spremi na stog
- c) učitaj vrijednost s vrha stoga u PC
- d) učitaj vrijednost s vrha stoga u SP

5. Navesti vrijednosti u registru PC **nakon** izvođenja instrukcija na adresama 1, 3, 7, 150.

1: MOV R0, \$1	PC = 2
2: MOV R1, \$1	
3: CALL \$100	PC = 100
4: ADD R1, R1, R0	
5: SUB R0, R1, R0	
6: CMP R1, 100	
7: BLT \$3 ; ako je R1 < 100 skoči na 3	PC = 3
...	
100: (ispis sadržaja u R0)	
...	
150: RET	PC = 4

6. Čemu služi operacijski sustav? Koje su njegove osnovne zadaće?

7. Što je to kontekst dretve? Od čega se sastoji? Tko s njime rukuje/upravlja?

8. Kada neka naprava dobavlja puno podataka koje treba pohraniti u radni spremnik, koji je najprikladniji način upravljanja s tom napravom? Obrazložiti.

1. (4 boda) U programskom brojilu nalazi se vrijednost **PC=1000**, u kazaljci stoga **SP=2000**. Na adresi 2000 (na vrhu stoga) nalazi se vrijednost **1200**. Što će se nalaziti u registrima PC i SP nakon izvođenja iduće instrukcije, ako se u memoriji na adresi 1000 nalazi instrukcija:

- | | | | | |
|----------------------|---------------------------|-----------|-------------------|----------------------------|
| a) B 1300 | (bezuovjetni skok) | rješenje: | PC= 1300 | SP= 2000 |
| b) CALL 1400 | (poziv potprograma) | rješenje: | PC= 1400 | SP= 1004 (ili 1001) |
| c) RET | (povratak iz potprograma) | rješenje: | PC= 1200 | SP= 1996 (ili 1999) |
| d) ADD R0, R1 | (zbroji R1 i R0 => R0) | rješenje: | PC= 1004/1 | SP= 2000 |

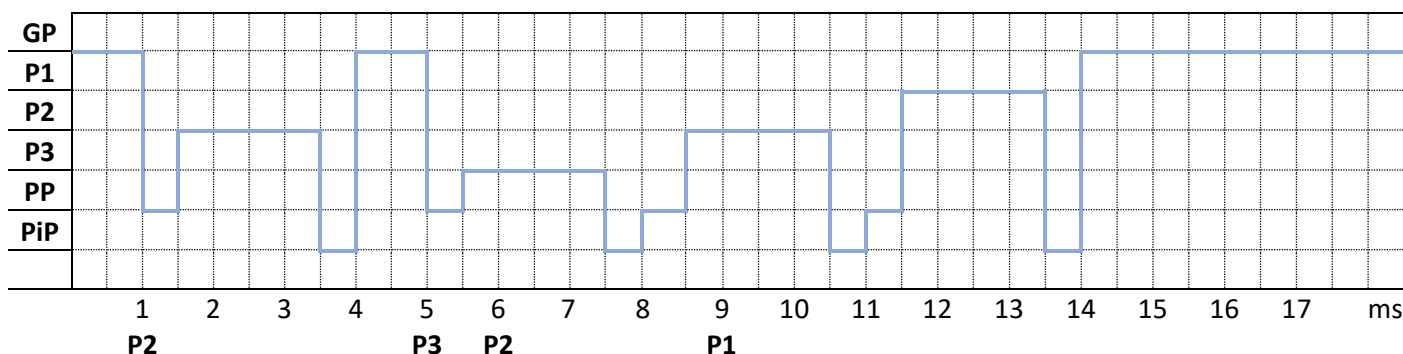
2. (1 bod) Pri zamjeni jedne dretve na procesoru drugom, potrebno je spremiti kontekst (sadržaje svih registara procesora) prve dretve te obnoviti kontekst druge dretve. Kako se inicira zamjena jedne dretve drugom (kako se zaustavlja prva da bi se obavile navedene operacije, kojim mehanizmom)?

Odgovor: **prekidom**

3. (1 bod) Pri prihvatu prekida **procesor** najprije zabranjuje daljnje prekidanje, prebacuje se u prekidni način rada i aktivira sustavsku kazaljku stoga, na taj stog pohranjuje PC i RS. Koju operaciju **procesor** još radi u tom postupku (koji je 4. korak u prihvatu prekida)?

Odgovor: **u PC stavlja adresu prekidnog potprograma**

4. (4 boda) U nekom sustavu dogodili su se prekidi P1-P3 prema slici (oznake ispod vremena, P2 u 1. ms; P3 u 5.; P2 u 6. te P1 u 9.). Pokazati stanje procesora do obrade svih navedenih prekida. Oznake: **GP**-izvođenje programa prije prekida, **PP**-prihvat prekida, **PiP**-povratak iz prekida, **Pi**-obrada prekida i (nakon prihvata). **PP** i **PiP** traju po 0,5 ms, a obrade Pi po 2 ms.



1. Što procesor sam radi pri prijemu prekida (do poziva prekidnog potprograma)?

Nadopuniti početak ispravnim operacijama (ne koristiti one koji nisu potrebne!), tj. staviti brojeke uz operacije.

1. procesor izvodi instrukciju neke dretve, prihvati prekida je omogućen

2. pojavljuje se zahtjev za prekid (prekidni signal)

___ na stog pohrani kazaljku stoga (SP) i podatkovni registar (PR)

___ na stog pohrani programsko brojilo (PC) i registar stanja (RS)

___ dovrši trenutnu instrukciju (regularno, ona se ne prekida)

___ prekini s izvođenjem trenutne instrukcije

___ prebaci se u korisnički način rada

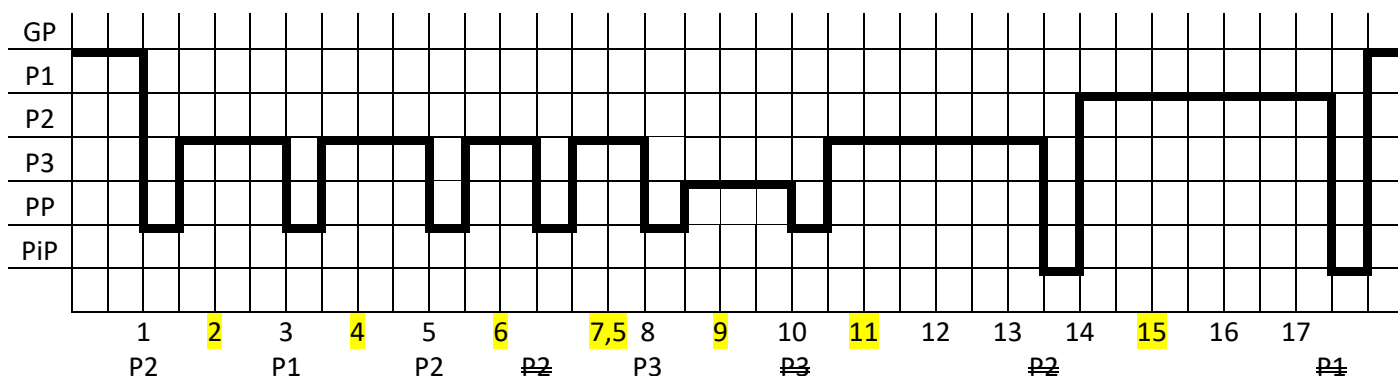
___ prebaci se u prekidni način rada

___ omogući daljnje prekidanje

___ zabrani daljnje prekidanje

X. u PC stavi adresu prekidnog potprograma (zadnja operacija)

2. U sustavu koji **nema sklop za prihvati prekida**, ali ima programsku potporu za prihvati prekida dogodili su se i obradili prekidi prema slici. Za označene trenutke opisati stanje varijabli TEKUĆI_PRIORITET i OZNAKA_ČEKANJA

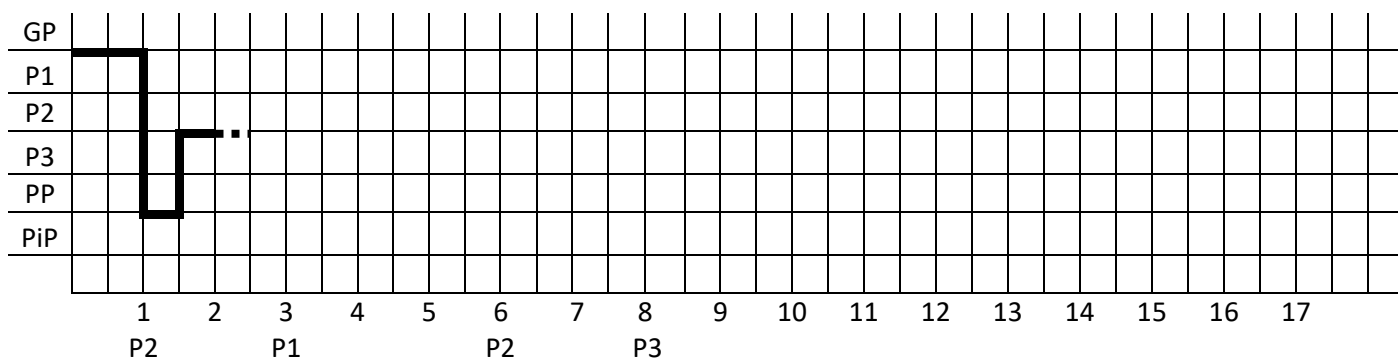


TEKUĆI_PRIORITET 2 ___ ___ ___ ___ ___ ___

OZNAKA_ČEKANJA[] 0 0 0 ___ ___ ___ ___ ___ ___

3 2 1

3. U sustavu **sa sklopom za prihvati prekida**, dogodili su se zahtjevi za prekidi prema slici. Pokazati kako će se obraditi ti zahtjevi, ako prihvati prekida (PP) traje 0.5 ms, povratak iz prekida 0.5 ms te obrade prekida po 3 ms.



1. (2) Što procesor sam radi pri prihvatu prekida (do poziva prekidnog potprograma)?

Nadopuniti početak ispravnim operacijama tj. staviti brojeke uz operacije (ne koristiti one se ne koriste).

1. procesor izvodi instrukciju neke dretve, prihvati prekida je omogućen
2. pojavljuje se zahtjev za prekid (prekidni signal)

___ omogući daljnje prekidanje

4 zabrani daljnje prekidanje

___ na stog pohrani kazaljku stoga (SP) i podatkovni registar (PR)

6 na stog pohrani programsko brojilo (PC) i registar stanja (RS)

3 dovrši trenutnu instrukciju (regularno, ona se ne prekida)

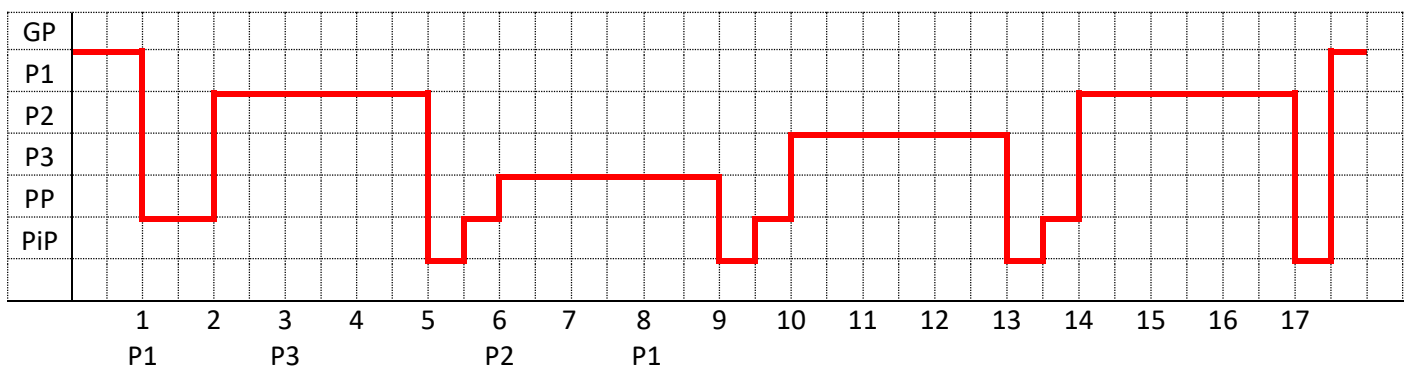
___ prekini s izvođenjem trenutne instrukcije

___ prebaci se u korisnički način rada

5 prebaci se u prekidni način rada

X. u PC stavi adresu prekidnog potprograma (zadnja operacija)

2. (4) U nekom sustavu dogodili su se prekidi prema slici. Prikazati stanja procesora dok se svi prekidi ne obrade. Prihvati prekida (PP) traje 1 ms, a povratak iz prekida (PP) traje 0,5 ms. Sama obrada (Pi) traje 3 ms.



3. (2) Kada ima smisla koristiti sklopove s izravnim pristupom spremniku (pristupni sklopovi koji od procesora traže sabirnicu i prenose podatak izravno u memoriju)?

- a) kad se radi o prioritetnoj napravi koju treba poslužiti prije svih ostalih
- b) kad se radi o najmanje prioritetnoj napravi koja se poslužuje nakon svih ostalih
- c) kada se nakon svakog prenesenog podatka on treba i koristiti
- d) kada se oni koriste tek nakon prijenosa bloka podataka

4. (2) Povežite opis s desne strane s pojmovima s lijeve.

- | | | | |
|---------|---|---|---|
| proces | 1 | 2 | niz instrukcija koje opisuju kako nešto napraviti |
| program | 2 | 3 | izvođenje instrukcija (uključujući stanje procesora, tj. sadržaje njegovih registara) |
| dretva | 3 | 1 | memorijski prostor u kojem se nalaze instrukcije, podaci i stog |

- Dretve A i B u dijelu svog koda mijenjaju varijablu brojilo operacijom $\text{brojilo} = \text{brojilo} + 1$.
 - Jesu li dretve A i B međusobno zavisne? DA NE
 - Treba li kod u kojem se brojilo mijenja biti kritični odsječak? DA NE
 - Ako dretve A i B paralelno izvode operaciju $\text{brojilo} = \text{brojilo} - 1$, a prethodna vrijednost je bila 10, nova vrijednost brojila (nakon što tu operaciju naprave obje dretve) može biti (zaokruži sve koje mogu biti):

-2-101278910111213
- Lamportov algoritam međusobnog isključivanja koristi varijable (polja) `ULAZ[ ]` i `BROJ[ ]`. Za dretvu I koja **se nalazi** u kritičnom odsječku vrijedi:
 - `BROJ[I] = 0`
 - `BROJ[I] = max(BROJ)` (ima najveću vrijednost)
 - `BROJ[I] = min(BROJ)` (ima najmanju vrijednost od brojeva koji imaju vrijednost veću od nule)
- Instrukcija `Ispitaj_i_postavi ZASTAVICA` (engl test-and-set: TAS ZASTAVICA) koristi dva uzastopna sabirnička ciklusa. Što ta instrukcija radi u drugom sabirničkom ciklusu?
 - čita vrijednost varijable ZASTAVICA
 - piše vrijednost 0 u varijablu ZASTAVICA
 - piše vrijednost 1 u varijablu ZASTAVICA
- Stanja dretvi u jednostavnom modelu jezgre su: aktivno, _____
- Sustav se početno nalazi u stanju S0. Redovi pripravnih dretvi i semafora uređeni su prema prioritetu – dretva većeg indeksa ima veći prioritet. Navesti stanja sustava nakon svakog idućeg poziva jezgrine funkcije. Pozivi su redom: 1. ČekajOSEM(4); 2. Zakasni(5); 3. ČekajBSEM(8); 4. ZapočniUI(5); 5. PostaviOSEM(4); 6. Otkucaj_sata(). Za prva dva poziva je već pokazano rješenje.

stanja	S0	S1	S2	S3	S4	S5	S6
Aktivna_D	5	4	3				
Pripravne_D	4 3 2 1	3 2 1	2 1				
Odgođene_D	-	-	4 ⁵				
UI[5]	-	-	-				
BSEM[8]	- (.v=1)	- (.v=1)	- (.v=1)				
OSEM[4]	- (.v=0)	5 (.v=0)	5 (.v=0)				

- Nadopuniti pseudokod proizvođača i potrošača da on bude primjenjiv na sustav s ograničenim međuspremnikom M kapaciteta N poruka.

```
proizvođač {
    ponavljaj {
```

```
        P = stvori poruku ()
```

```
        M[ULAZ] = P
```

```
        ULAZ = (ULAZ + 1) MOD N
```

```
        Postavi_OSEM(1)
```

```
    }
    do zauvijek
```

```
}
```

```
potrošač {
    ponavljaj {
```

```
        Čekaj_OSEM(1)
```

```
        R = M[IZLAZ]
```

```
        IZLAZ = (IZLAZ + 1) MOD N
```

```
        obradi poruku (R)
```

```
    }
    do zauvijek
```

```
}
```

7. (1) Jezgrine funkcije pozivaju se:
- Lamportovim algoritmom
 - prekidima
 - instrukcijom CALL
 - instrukcijom KERNEL
8. (1) Osnovna četiri stanja dretvi u jednostavnom modelu jezgre su:
- aktivno
 - pasivno
 - _____
 - _____
9. (1) Ako se u redu nekog semafora nalazi nekoliko blokiranih dretvi, stanje semafora je:
- prolazno
 - neprolazno
10. (1) Što će napraviti poziv PostaviOSEM(1) ako se u redu tog semafora ne nalazi niti jedna dretva?
- _____
11. (1) Neka dretva pozove Započni_UI(2). Što će se dogoditi s tom dretvom u tom pozivu?
- _____
12. (1) U redu Odgođene_D nalaze se dretve D1, D3 i D2 tim redom s pripadnim vrijednostima Zadano_kašnjenje: 3, 5 i 1. Koliko otkucaja sata treba čekati dretva D2 prije nego li će nastaviti s radom (biti odblokirana iz tog reda)?
- _____
13. (3) Jedan proizvođač i jedan potrošač komuniciraju preko ograničenog međuspremnika M kapaciteta 10 poruka korištenjem slijedećeg pseudokoda.

<pre> proizvođač { ponavljaj { P = stvori poruku () ČekajOSEM(2) M[ULAZ] = P ULAZ = (ULAZ + 1) MOD N Postavi_OSEM(1) } do zauvijek } </pre>	<pre> potrošač { ponavljaj { Čekaj_OSEM(1) R = M[IZLAZ] IZLAZ = (IZLAZ + 1) MOD N PostaviOSEM(2) obradi poruku (R) } do zauvijek } </pre>
---	---

U nekom trenutku varijable imaju vrijednosti: ULAZ=8, IZLAZ=5.

M[]	0	1	2	3	4	5	6	7	8	9
						IZLAZ			ULAZ	

Kolike su vrijednosti semafora 1 i 2 u tom trenutku?

OSEM[1].v = ____ OSEM[2].v = ____

- dretve uzimaju redom prispijeća i izvode do završetka (a tek onda se uzima druga dretva s početka reda)
- dretve uzimaju prema prioritetu i izvode do završetka (a tek onda se uzima druga dretva prema prioritetu)
- dretve uzimaju redom prispijeća, izvode kvant vremena T_q te vraćaju na kraj reda (a tek onda se uzima druga dretva s početka reda)

- a) prema redu prispjeća
- b) prema prioritetu
- c) podjelom vremena

- a) red prispjeća i podjela vremena
- b) red prispjeća i prioritet
- c) prioritet i podjela vremena

[illegible][illegible]

S3 ZapočniUI(1) **S1** Odgodi(5) **S2** ČekajOSEM(1) **S0** ČekajOSEM(2)
PrekidUI(1) **S5** Otkucaj_sata() PostaviOSEM(1) **S4** PostaviOSEM(2)

20. (1) Raspoređivanje dretvi u višeprocorskom sustavu (zaokružiti) **koristi** / ne koristi iste kriterije kao i za jednoprocesorske sustave (prioritet, podjela vremena, red prispjeća i njihove kombinacije).

21. Kod upravljanja spremnikom straničenjem procesor mora imati sklop za upravljanje spremnikom. Sklop koristi tablicu prevođenja da bi preveo logičku adresu u fizičku.

(a) Od čega se sastoji svaki zapis (redak) u tablici prevođenja?

adrese okvira u kojem se stranica nalazi

zastavica (među kojima je i bit prisutnosti)

(b) Neka stranica čiji se podatak želi pročitati nije u radnom spremniku (već na disku). Što će sklop napraviti u tom slučaju pokušaja dohvata podatka iz te stranice?

izazvat će prekid

22. Jedinica podataka na disku je sektor. Međutim, kod organizacije podataka na disku u okviru datotečnih sustava, koristi se veća jedinica. Koja?

(a) oktet (bajt, B)

(b) staza (svi sektori na jednoj ploči jednako udaljeni od središta rotacije)

(c) cilindar (sve staze na svim pločama jednako udaljene od središta rotacije)

(d) blok (skup od nekoliko uzastopnih sektora)

23. Nabrojite elemente opisnika datoteke.

ime, direktorij, datum zadnje promjene, vlasnik, prava pristupa, opis smještaja na disku

24. Ako se raspolaže s 10 diskova od 4 TB, koji **korisni** kapacitet (za podatke tj. datoteke) će se dobiti ako se oni spoje korištenjem načina:

RAID 0 => **$10 * 4 = 40 \text{ TB}$**

RAID 1 => **$10/2 * 4 = 20 \text{ TB}$**

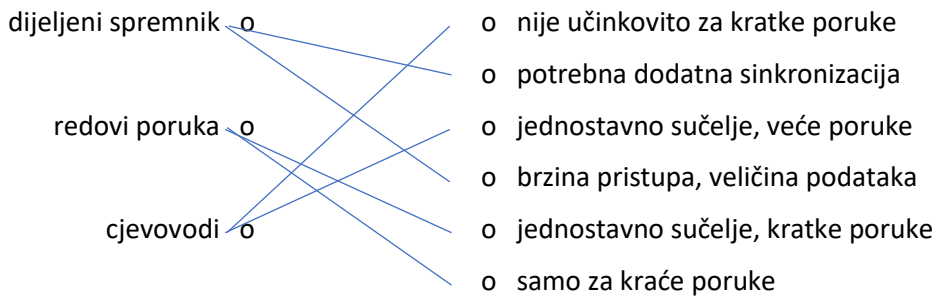
RAID 5 => **$(10-1) * 4 = 36 \text{ TB}$**

RAID 6 => **$(10-2) * 4 = 32 \text{ TB}$**

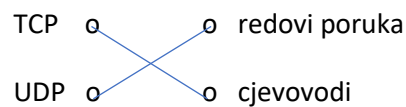
25. Jedan RAID 5 sustav se sastoji od 5 diskova, a drugi RAID 5 od 10 istovrsnih diskova. Koji sustav je pouzdaniji, tj. ima očekivano dulje vrijeme rada bez gubitaka podataka (bez zamjene diskova koji će se s vremenom pokvariti), onaj s 5 ili onaj s 10 diskova? Zašto?

RAID 5 s 5 diskova – manje diskova = manja vjerojatnost dvostruka kvara

5. Povezati načine međudretvene komunikacije unutar jednog računalnog sustava s njihovim svojstvima (svaki s 2).



6. Komunikacija protokolima TCP i UDP u raspodijeljenom sustavu je slična komunikaciji korištenjem reda poruka i cjevovoda unutar jednog računalnog sustava. Povezati protokole koji koriste ista načela komunikacije.



7. Virtualizacijom (pokretanjem više virtualnih računala na istom fizičkom računalu) dobiva se:

- a) povećanje procesne moći računala
- b) moguće bolje iskorištenje računala (npr. poslužiteljskih)
- c) izolacija pojedinih usluga (koje se pokreću u različitim virtualnim računalima)
- d) mogućnost simulacije nepostojećeg sklopovlja

8. Održavanje računala (neke tvrtke/ustanove/kućnih korisnika) koje uključuje redovito ažuriranje operacijskog sustava, periodičke izrade sigurnosne kopije podataka (backup) te korištenje i redovito ažuriranje antivirusne zaštite

- a) pomaže u zaštiti korisničkih podataka od zlonamjernih pokušaja dohvata
- b) smanjuje zauzeće spremničkog i diskovnog prostora
- c) skraćuje vrijeme obnove sustava u slučaju kvara
- d) povećava performanse sustava

1. (1) Kod straničenja sklop pretvara _____ adresu u _____ adresu.
2. (1) Što opisuje svaki redak tablice prevođenja (kod straničenja)?

3. (1) U tablici prevođenja postoji redak 75 u kojem je vrijednost 55 i bit prisutnosti=1. Kako će se prevesti adresa {stranica, odmak} = {75, 44}?

4. (1) Što je to promašaj u kontekstu straničenja?

5. (1) Jedinica podataka na disku je sektor. Kako se jedinstveno definira jedan sektor – što sve treba (od geometrije diska) da bi se došlo do „adrese“ sektora na disku?

6. (1) Navesti nekoliko elemenata (bar tri) koji se nalaze u opisniku datoteke.

7. (1) Zašto se diskovi spajaju u RAID? Koje se sve prednosti time mogu dobiti (nekim konfiguracijama)?

8. (1) Ako na raspolaganju stoji 10 diskova, hoće li se veći korisni kapacitet dobiti s RAID1, RAID5 ili RAID6?

9. (1) Koje se prednosti u nekim situacijama mogu ostvariti korištenjem virtualizacije?

